

m-0

---

## VOLUME 2



---

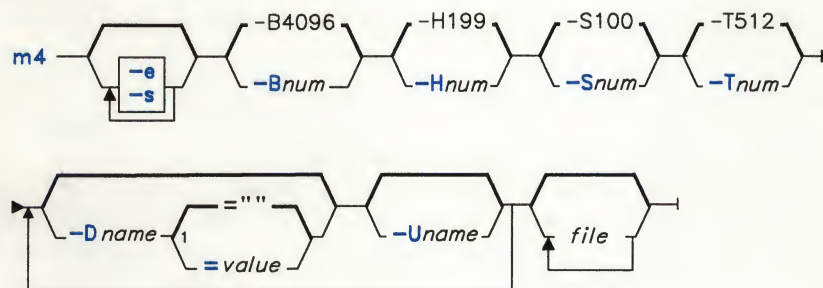
## m4

---

### Purpose

Preprocesses files, expanding macro definitions.

### Syntax



OL805434

### Description

The **m4** command is a macroprocessor used as a preprocessor for C and other languages. You can use it to process built-in macros or user-defined macros. Each *file* is processed in order. If you do not specify a *file* or if you give a minus (-) as a file name, **m4** reads standard input. It writes the processed macros to standard output. Macro calls follow the form:

```
macroname(argument . . . )
```

The left parenthesis must immediately follow macroname. If the left parenthesis does not follow the name of a defined macro, **m4** reads it as a macro call with no arguments. Macro names consist of ASCII alphabetic letters, digits, and the underscore character (\_). Extended characters are not allowed in macro names. The first character cannot be a digit.

While collecting arguments, **m4** ignores unquoted leading blanks, tabs, and new-line characters. Use single quotation marks to quote strings. The value of a quoted string is the string with the quotation marks stripped off.



When **m4** recognizes a macro, it collects arguments by searching for a matching right parenthesis. If you supply fewer arguments than appear in the macro definition, **m4** considers the trailing arguments in the definition to be null. Macro evaluation proceeds normally during the collection of the arguments. All commas or right parentheses within the value of a nested call are translated literally; they do not need an escape character or quotation marks. After collecting arguments, **m4** pushes the value of the macro back onto the input stream and scans again.

## Built-in Macros

The **m4** command makes available the following built-in macros. You may redefine them, but you will lose the original meaning. The values of these macros are null unless otherwise stated:

<b>define</b> ( <i>name</i> , <i>new_name</i> )	Replaces the macro <i>name</i> with the value of <i>new_name</i> . The <i>new_name</i> string can take the form <i>\$n . . .</i> (where <i>n</i> is a digit). In this case, each occurrence of <i>n</i> in the replacement text is replaced by the <i>n</i> -th argument of <i>name</i> . <i>\$0</i> is the name of the macro. The null string replaces missing arguments. The number of arguments replaces <i>\$#</i> . A comma-separated list of all arguments replaces <i>\$*</i> . <i>\$@</i> acts like <i>\$*</i> , but each argument is quoted with the current quotation character (see <b>changequote</b> ).
<b>undefine</b> ( <i>name</i> )	Removes the definition of <i>name</i> .
<b>defn</b> ( <i>name</i> . . . )	Returns the quoted definition of <i>name</i> .
<b>pushdef</b> ( <i>name</i> , <i>new_name</i> )	Redefines <i>name</i> with <i>new_name</i> as in <i>define</i> , but save any previous definition.
<b>popdef</b> ( <i>name</i> . . . )	Removes the current definition of <i>name</i> and returns to the previous definition, if one existed.
<b>ifdef</b> ( <i>name</i> , <i>true</i> ,[ <i>false</i> ])	Returns the value of <i>true</i> only if <i>name</i> is defined, otherwise return <i>false</i> . If you do not supply <i>false</i> , its value is null. <b>Note:</b> The word <b>unix</b> is predefined.
<b>shift</b> ( <i>argument</i> . . . )	Returns all but the first argument. The other arguments are quoted and pushed back with commas in between. The quoting nullifies the effect of the extra scan that will subsequently be performed.
<b>changequote</b> ( <i>L</i> , <i>R</i> )	Changes quote symbols to <i>L</i> and <i>R</i> . The symbols can be up to 5 bytes long. <b>changequote</b> without arguments restores the original values ( ' ' ).

<b>changecom</b> ( <i>Lcom</i> , <i>Rcom</i> )	Changes left and right comment markers from the default # and new-line character to <i>Lcom</i> and <i>Rcom</i> . With no arguments, the comment mechanism is disabled. With one argument, the left marker becomes the parameter and the right marker becomes a new-line character. With two arguments, both markers are affected. Comment markers can be up to 5 bytes long.
<b>divert</b> ( <i>num</i> )	Changes the current output stream to stream <i>num</i> . There are 10 output streams, numbered 0-9. The final output is the concatenation of the streams in numerical order. Initially, stream 0 is the current stream. <b>m4</b> discards output diverted to a stream other than 0-9.
<b>undivert</b> ( <i>num</i> . . . )	Causes immediate output of text from the specified diversions (or all diversions if there is no argument). Text may be undiverted into another diversion. Undiverting discards the diverted text.
<b>divnum</b>	Returns the value of the current output stream.
<b>dnl</b>	Reads and discards characters up to and including the next new-line character.
<b>ifelse</b> ([ <i>string1</i> , <i>string2</i> , <i>true</i> , <i>false</i> ] . . . )	If <i>string1</i> and <i>string2</i> are the same, then the value is <i>true</i> . If they are not and if there are more than four arguments, <b>m4</b> repeats the process with the additional arguments (4, 5, 6, and 7). Otherwise, the value is either <i>false</i> or null if you provide no value for <i>false</i> .
<b>incr</b> ( <i>num</i> )	Returns the value of its argument incremented by 1.
<b>decr</b> ( <i>num</i> )	Returns the value of its argument decreased by 1.
<b>eval</b> ( <i>expr</i> [, <i>num1</i> [, <i>num2</i> ]])	Evaluates its first argument as an arithmetic expression, using 32-bit arithmetic. The operators you can use include +, -, *, /, %, ^ (exponentiation), bitwise &,  , ~, and ^ relationals, and parentheses. Octal and hex numbers can be specified as in C. <i>num1</i> specifies the radix for the result of the expression. The default radix is 10. The optional <i>num2</i> specifies the minimum number of digits in the result.
<b>len</b> ( <i>string</i> )	Returns the number of bytes in <i>string</i> .
<b>dlen</b> ( <i>string</i> )	Returns the number of displayable characters in <i>string</i> ; that is, 2-byte extended characters are counted as one displayable character.



<b>index</b> ( <i>s1,s2</i> )	Returns the position in the string <i>s1</i> where the string <i>s2</i> begins (zero origin), or -1 if the second parameter does not occur.
<b>substr</b> ( <i>string,position,num</i> )	Returns a substring of <i>string</i> . The beginning of the substring is selected with <i>position</i> , and <i>num</i> indicates the length of the substring. Without <i>num</i> , the substring includes everything to the end of the first string.
<b>translit</b> ( <i>string,from,to</i> )	Transliterates the characters in <i>string</i> from the set given by <i>from</i> to the set given by <i>to</i> . No abbreviations are permitted. Two-byte extended characters are correctly mapped into the corresponding replacement characters.
<b>include</b> ( <i>file</i> )	Returns the contents of <i>file</i> or displays an error message if it cannot access the file.
<b>sinclude</b> ( <i>file</i> )	Returns the contents of <i>file</i> , but it gives no error message if <i>file</i> is inaccessible.
<b>syscmd</b> ( <i>command</i> )	Runs the AIX <i>command</i> . No value is returned.
<b>sysval</b>	Returns the return code from the last call to <i>syscmd</i> .
<b>maketemp</b> ( . . . XXXXX . . . )	Replaces XXXXX in its argument with the current process ID number.
<b>m4exit</b> ( <i>value</i> )	Exits from <b>m4</b> immediately, returning the specified exit <i>value</i> (the default is 0).
<b>m4wrap</b> ( <i>lastmacro</i> )	Runs <i>lastmacro</i> after reading the end-of-file character. For example: <code>m4wrap('cleanup()')</code> runs the <code>cleanup</code> macro at the end of <b>m4</b> .
<b>errprint</b> ( <i>message</i> )	Includes <i>message</i> on the diagnostic output file.
<b>dumpdef</b> ([ <i>name</i> . . . ])	Writes to standard output the current names and definitions for the named items or for all if no arguments are provided.
<b>traceon</b> ( <i>macro</i> )	Turns on tracing for <i>macro</i> . If none is named, tracing is turned on for all macros.
<b>traceoff</b> ( <i>macro</i> . . . )	Turns off trace globally and for any <i>macro</i> specified. Macros specifically traced by <b>traceon</b> can be untraced only by specific calls to <i>traceoff</i> .

## Flags

- Bnum**            Makes *num* the size of the push-back and parameter collection buffers (the default is 4096).
  - e**                Operates interactively. Interrupts are ignored and the output is not buffered.
  - Hnum**            Makes *num* the size of the symbol table hash array (the default is 199). The size must be a prime number.
  - s**                Enables the line sync output for the C preprocessor (#line . . . ).
  - Snum**            Makes *num* the size of the call stack (the default is 800 slots). Macros take three slots, and nonmacro arguments take one.
  - Tnum**            Makes *num* the size of the token buffer (the default is 512 bytes).
- The preceding flags must appear before any file names and before any **-D** or **-U** flags.
- Dname[=val]**    Define *name* as *val*. If *val* is not specified, *name* becomes null.
  - Uname**            Undefines a *name* previously defined with the **-D** flag.

## Example

To preprocess a C language program with **m4** and compile it:

```
m4 prog.m4 >prog.c
cc prog.c
```

## Related Information

The following commands: “**cc**” on page 140 and “**cpp**” on page 210.

“Overview of International Character Support” in *Managing the AIX Operating System*.

# mail, Mail

---

## mail, Mail

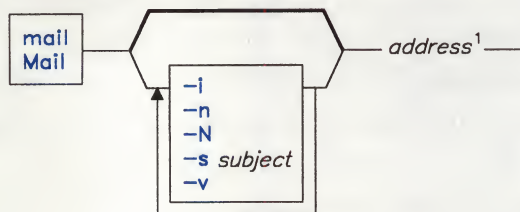
---

### Purpose

Sends and receives mail.

### Syntax

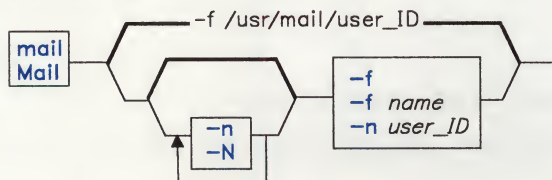
#### Sending Mail:



<sup>1</sup>See `sendmail` for address formats.

AJ2FL262

#### Handling Mail:



AJ2FL260

### Description

The **mail** program allows you to:

- Compose a message and send it
- Receive a message and look at it
- Store received messages in your mailbox, or in folders
- Discard messages.



To send a message to one or more persons, enter **mail** on the command line with arguments that are the network addresses of the people to receive the message. When **mail** starts, you can type the message using an editor similar to **ed**. When you are finished with the message, press the **Enter** key at the end of a line, and use a **Ctrl-D** (EOF) sequence at the beginning of the next line to exit the editor and send the message.

When you have messages in your mailbox, the system displays a message to tell you. The default message is:

```
[YOU HAVE NEW MAIL]
```

To look at the contents of your mailbox, enter the **mail** command without arguments on the command line. The program displays a listing of the messages in your mailbox and allows you to look at them, reply to them or dispose of them.

## Reading Incoming Mail

To receive and read incoming mail, use **mail** with no arguments:

```
mail
```

The **mail** command then checks your system mailbox (**/usr/mail/user-id**) and displays a one-line entry for each message in the system mailbox similar to:

```
"/usr/mail/geo": 2 messages 2 new
>N 1 amy      Thu Sep 17 14:36 13/359 "Dept Meeting"
  N 2 amy      Thu Sep 17 16:28 13/416 "Dept Meeting Delayed"
&
```

The **>** symbol indicates the *current message*, or the message that commands act on if you do not specify a message number or list of message numbers. The other fields, in order, in the listing represent:

1. Message number
2. User address of the sender
3. Date the message was sent
4. Size of the message in lines/characters
5. The subject of the message (if one was included in the message).

From the **mail** command prompt **&**, you can enter commands to look at, reply to, save, discard, or otherwise manage the contents of the mailbox. To display a summary of some of the commands that you can use to handle mail in your mailbox, enter **?** at the **mail** command prompt. For more information on those commands and information on additional commands, refer to Figure 3 on page 611.

Many mailbox commands allow you to specify groups of messages upon which to perform the command. Commands that allow groups of messages use the parameter *msg\_lst* in the command format. For example, the format of the **f** command (display information about messages) appears as:

```
& f msg_lst
```

## mail, Mail

---

In this format *msg\_lst* can be one of the following:

- One or more message numbers separated by spaces

& f 1 2 4 7

- A range of message numbers indicated by the first and last numbers in the range separated by a dash

& f 2-5

is the same as,

& f 2 3 4 5

- One or more addresses separated by spaces to apply the command to messages received from those addresses,

& f amy geo@zeus

The characters entered for an address need not match the address exactly. They must only be contained in the address field of the messages in either upper- or lower-case. Therefore, the request for address amy matches all of the following addresses (and many others):

- amy
- AmY
- amy@zeus
- hamy

- A string, preceded by a slash, to match against the **Subject:** field of the messages,

& f /meet

applies the command to all messages whose **Subject:** field contains the letters meet in upper- or lower-case. The characters entered for a match pattern do not need to match the **Subject:** field exactly. They must only be contained in the Subject: field of the messages in either upper- or lower-case. Therefore, the request for subject meet matches all of the following subjects (and many others):

- Meeting on Thursday
- Come to meeting tomorrow
- MEET ME IN ST. LOUIS

The special character \* (asterisk) addresses all messages, ^ (caret) addresses the first message, and \$ (dollar sign) addresses the last message.



The following table lists the **mail** commands and describes their functions.

Command	Function
=	Echoes the number of the current message.
#	Comment character for writing comments in mail script files.
-n	Goes to the previous message and displays it. If given a number argument of <i>n</i> , goes to the <i>n</i> th previous message and displays it.
?	Displays a brief summary of commands.
!sh_cmd	Executes the AIX shell command specified by <i>sh_cmd</i> .
alias	(a) With no arguments, displays all currently defined aliases. With one argument, displays alias. With more than one argument, creates a new or changes an old alias.
alternates alt_list	(alt) The <b>alternates</b> command is useful if you have accounts on several machines. Use it to inform <b>mail</b> that the addresses listed in <i>alt_list</i> all refer to you. Then, when you <b>reply</b> to messages, <b>mail</b> does not send a copy of the message to any of the addresses given in <i>alt_list</i> . If you enter the <b>alternates</b> command with no argument, <b>mail</b> displays the current set of alternate names.
chdir dir	(cd) Changes your working directory to the directory <i>dir</i> . If no directory is given, it changes to your login directory.
copy msg_lst file	(c, co) Appends each message in <i>msg_lst</i> in turn to the end of <i>file</i> . Displays the file name in quotes, followed by the line count and character count, on the user's terminal. Does not delete any messages when you quit.
delete msg_lst	(d) Marks the messages in <i>msg_lst</i> to be deleted when you quit <b>mail</b> . Deleted messages are not saved in <b>mbox</b> , nor are they available for most other commands. However, you can restore messages that you have deleted while in the same mailbox session (see the <b>undelete</b> mailbox command).

Figure 3 (Part 1 of 6). Mailbox Commands

## mail, Mail

---

Command	Function
<b>discard</b> [ <i>fld_lst</i> ]	(di) Identical to the <b>ignore</b> command.
<b>dp</b>	Deletes the current message and displays the next message. If there is no next message, <b>mail</b> displays the message at EOF.
<b>dt</b>	Identical to the <b>dp</b> command.
<b>echo</b> <i>string</i>	Displays the character string <i>string</i> on the command line.
<b>edit</b> <i>msg</i>	(e) Activates the editor that you define with the <b>set EDITOR=</b> statement and loads message <i>msg</i> into the editor. When you exit the editor, the saved message is replaced in the mailbox being processed.
<b>exit</b>	(ex or x) Exits to the shell without changing the mailbox being processed. The mailbox returns to the condition that it was when <b>mail</b> was started. Messages marked to be deleted are not deleted.
<b>file</b> [ <i>name</i> ]	(fi) Identical to the <b>folder</b> command.
<b>folder</b> [ <i>name</i> ]	(fo) Switches to a new mail file or folder. With no arguments, displays the name of the mailbox that you are currently reading. If an argument is included, it stores the current mailbox with changes (such as messages deleted) and reads in the new mailbox specified by the <i>name</i> parameter. Some special conventions are recognized for the <i>name</i> : <ul style="list-style-type: none"><li>• # refers to the previous file</li><li>• % refers to the system mailbox</li><li>• &amp; refers to your personal mailbox (\$HOME/<b>mbox</b>)</li><li>• <i>name</i> refers to a file in your folder directory.</li></ul>
<b>folders</b>	Lists the names of the folders in your folder directory.
<b>from</b> <i>msg_lst</i>	(f) Displays the headings of messages in <i>msg_lst</i> .
<b>group</b>	(g) Identical to the <b>alias</b> command.

Figure 3 (Part 2 of 6). Mailbox Commands



Command	Function
<b>headers</b>	(h) Lists the headings in the current group of messages (each group of messages contains 20 messages by default; change this with the <b>set screen = statement</b> ).
<b>help</b>	Identical to question mark (?).
<b>hold</b> <i>msg_lst</i>	(ho) Marks each message in <i>msg_lst</i> to be saved in your system mailbox instead of in <b>mbox</b> . Does not override the <b>delete</b> command.
<b>if</b> <i>condition</i> <b>else</b> <b>endif</b>	Construction for conditional execution of <b>mail</b> commands. Commands following <b>if</b> are executed if <i>condition</i> is true. Commands following <b>else</b> are executed if <i>condition</i> is not true. The <b>else</b> is not required. The <b>endif</b> ends the construction and is required. The <i>condition</i> can be receive (receiving mail) or send (sending mail).
<b>ignore</b> [ <i>fld_lst</i> ]	Adds the header fields in <i>fld_lst</i> to the list of fields to be ignored. Ignored fields are not displayed when you look at a message with the <b>t</b> or <b>p</b> commands. Use this command to suppress machine-generated header fields. Use the <b>Type</b> and <b>Print</b> commands to print a message in its entirety, including ignored fields. If <b>ignore</b> is executed with no arguments, it lists the current set of ignored fields.
<b>list</b>	(l) Displays a list of valid <b>mail</b> commands.
<b>local</b>	Lists other names for the local host.
<b>mail</b> <i>addr_lst</i>	(m) Activates the mail editor to allow you to create and send a message to people specified in <i>addr_lst</i> .
<b>mbox</b> <i>msg_lst</i>	Indicates that the messages in <i>msg_lst</i> be sent to your personal mailbox when you quit. This operation is the default action for messages that you have looked at if you are looking at your system mailbox and the <b>hold</b> option is not set.
<b>more</b> <i>msg_lst</i>	(mo) Displays the messages in <i>msg_lst</i> using the defined pager program to control display to the screen.
<b>More</b> <i>msg_lst</i>	(Mo) Like <b>more</b> but also displays ignored header fields. See <b>more</b> and <b>ignore</b> .

Figure 3 (Part 3 of 6). Mailbox Commands



## mail, Mail

---

Command	Function
<b>new</b> <i>msg_lst</i>	Identical to the <b>unread</b> command.
<b>New</b> <i>msg_lst</i>	Identical to the <b>Unread</b> command.
<b>next</b> [ <i>msg</i> ]	(n) Makes the next message in the mailbox the current message and displays that message. With an argument list, it displays the next matching message.
<b>page</b> <i>msg_lst</i>	(pa) Identical to the <b>more</b> command.
<b>Page</b> <i>msg_lst</i>	(Pa) Identical to the <b>More</b> command.
<b>preserve</b>	(pre) Identical to the <b>hold</b> command.
<b>print</b> <i>msg_lst</i>	(p) Displays the messages in <i>msg_lst</i> .
<b>Print</b> <i>msg_lst</i>	(P) Like <b>print</b> but also displays ignored header fields. See <b>print</b> and <b>ignore</b> .
<b>quit</b>	(q) Ends the session and returns to the shell. Before ending, <b>mail</b> saves all messages that have not been deleted or saved in your personal mailbox ( <b>\$HOME/mbox</b> ). It keeps all messages marked with <b>hold</b> or <b>preserve</b> and those messages that have not been looked at, in the system mailbox. It removes all other messages from the system mailbox. If given while editing a mailbox file with the <b>-f</b> flag, then the edit file is saved with changes. If the edit file cannot be saved, <b>mail</b> does not exit. Use the <b>exit</b> command to exit without saving the changes.
<b>reply</b> <i>msg</i>	Allows you to create and send mail to the people who sent and received the message specified in <i>msg</i> .
<b>Reply</b> <i>msg</i>	Allows you to create and send mail only to the person who sent the message specified in <i>msg</i> .
<b>respond</b> <i>msg</i>	Identical to the <b>reply</b> command.
<b>Respond</b> <i>msg</i>	Identical to the <b>Reply</b> command.

Figure 3 (Part 4 of 6). Mailbox Commands

Command	Function
<b>retain</b> [ <i>fld_lst</i> ]	Adds the header fields in <i>fld_lst</i> to the list of fields to be retained. Retained fields are displayed when you look at a message with the <b>t</b> or <b>p</b> commands. Use this command to define which header fields you want displayed. Use the <b>Type</b> and <b>Print</b> commands to print a message in its entirety, including fields that are not retained. If <b>retain</b> is executed with no arguments, it lists the current set of retained fields.
<b>save</b> <i>msg_lst file</i>	(s) Appends the messages specified in <i>msg_lst</i> to <i>file</i> . Displays the file name and the size of the file when the operation is complete. If you save a message to a file, that message is not returned to the system mailbox nor saved in your personal mailbox when you quit the <b>mail</b> program.
<b>set</b> [ <i>option</i> ]	(se) With no arguments, prints all variable values. Otherwise, sets an option as specified in <i>option</i> . The <i>option</i> field can be either the name of a <b>binary</b> option (an option that is either set or not set) or a statement of the form:  option=value  That assigns a value to a valued option. Binary and valued options are described later in this command description.
<b>shell</b>	(sh) Invokes an interactive version of the shell.
<b>size</b> <i>msg_lst</i>	Displays the sizes in lines/characters of the messages in <i>msg_lst</i> .
<b>source</b> <i>file</i>	(so) Reads <b>mail</b> commands from <i>file</i> .
<b>top</b> <i>msg_lst</i>	Displays the top few lines of the messages specified by <i>msg_lst</i> . The number of lines displayed is determined by the valued option <b>toplines</b> and defaults to five.
<b>touch</b> <i>msg_lst</i>	When operating with your system mailbox, this command marks the messages in <i>msg_lst</i> to be moved to your personal mailbox when you quit the <b>mail</b> program, even though you have not read the listed messages. The messages appear in your personal mailbox as unread messages. When you use <b>touch</b> , the last message in <i>msg_lst</i> becomes the current message.

Figure 3 (Part 5 of 6). Mailbox Commands



## mail, Mail

---

Command	Function
<code>type msg_lst</code>	(t) Identical to the <b>print</b> command.
<b>Type</b> <code>msg_lst</code>	(T) Identical to the <b>Print</b> command.
<b>unalias</b> <code>al_lst</code>	Removes the defined aliases specified in <code>al_lst</code> .
<b>undelete</b> <code>msg_lst</code>	(u) Removes the messages in <code>msg_lst</code> from the list of messages to be deleted when you quit <b>mail</b> .
<b>unread</b> <code>msg_lst</code>	(U) Marks each message in <code>msg_lst</code> as <i>not</i> having been read.
<b>Unread</b> <code>msg_lst</code>	Identical to the <b>unread</b> command.
<b>unset</b> <code>option_lst</code>	Discards the values of the options specified in <code>option_lst</code> . This action is the inverse of the <b>set</b> command.
<b>version</b>	(ve) Displays the version banner for the <b>mail</b> program.
<b>visual</b> <code>msg</code>	(v) Activates the editor that you define with the <code>set VISUAL=</code> statement and loads message <code>msg</code> into the editor. When you exit the editor, the saved message is replaced in the mailbox being processed.
<b>write</b> <code>msg_lst file</code>	(w) Appends the messages specified in <code>msg_lst</code> to <code>file</code> . Displays the file name and the size of the file when the operation is complete. Does not include message headers in the file.
<b>xit</b>	(x) Identical to the <b>exit</b> command.
<b>z</b> [ <b>+</b> ] [ <b>-</b> ]	Changes the current message group (group of 20 messages) and displays the headings of the messages in that group. If a <b>+</b> or no argument is give, then headings in the next group are shown. If a <b>-</b> argument is given, the headings in the previous group are shown.

Figure 3 (Part 6 of 6). Mailbox Commands

## Handling Outgoing Mail

To compose and send a message, use **mail** with the following format:

**mail** `addr_lst`

In this format `addr_lst` is a list of user addresses separated by spaces. This command activates the mail editor so that you can compose a message to be sent to the specified addresses.

By default, **mail** treats lines beginning with the character `~` (tilde) as special while you are composing a message. For instance, typing `~m` on a line by itself places a copy of the current message into the response, shifting it to the right by one tab stop.

Other escapes set up subject fields, add and delete recipients of the message, and allow the user to escape to an editor to revise the message, or to a shell to run other commands. You can change the escape character to something other than a tilde with the **set escape=** statement. To view a summary of many useful commands, enter `~?` on a line by itself while in the mail editor.

Figure 4 shows a summary of the mail editor commands. Use these commands only while in the mail editor. The editor recognizes commands only if you enter them at the beginning of a new line.

Command	Function
<code>~!cmd</code>	Executes the shell command, <i>cmd</i> and returns to the message.
<code>~b addr_lst</code>	Adds names in <i>addr_lst</i> to the list of people to receive blind copies of the message.
<code>~c addr_lst</code>	Adds names in <i>addr_lst</i> to the list of people to receive copies of the message.
<code>~d</code>	Reads the file <b>dead.letter</b> from your home directory into the message.
<code>~e</code>	Activates the editor that you have specified with the <b>set EDITOR=</b> statement using the message text in the current message. When you exit that editor, you return to the mail editor to continue appending the changed message, or to send the message by exiting the <b>mail</b> program.
<code>~f msg_lst</code>	Reads the named messages into the message being sent. If no messages are specified, reads the current message. This command works only if you entered the mail editor from the mailbox listing using the <b>m</b> or <b>r</b> mailbox commands.
<code>~h</code>	Allows you to edit the message header fields by typing each one in turn. Allows you to append text to the end or modify the field using the current terminal erase and kill characters.

Figure 4 (Part 1 of 2). Mail Editor Commands



## mail, Mail

---

Command	Function
<code>~m msg_lst</code>	Reads the named messages into the message being sent, shifted right one tab. If no messages are specified, reads the current message. This command works only if you entered the mail editor from the mailbox listing using the <b>m</b> or <b>r</b> mailbox commands.
<code>~p</code>	Displays the message as it currently exists, prefaced by the message header fields.
<code>~q</code>	Aborts the message being created without sending it. Saves the message in <b>dead.letter</b> in your home directory if the <b>save</b> option is set.
<code>~r filename</code>	Reads the named file into the message.
<code>~s string</code>	Changes the <b>Subject:</b> field to the phrase specified in <i>string</i> .
<code>~t addr_lst</code>	Adds the addresses in <i>addr_lst</i> to the <b>To:</b> field of the message.
<code>~v</code>	Activates the editor that you have specified with the <code>set VISUAL=</code> statement using the message text in the current message. When you exit that editor, you return to the mail editor to continue appending to the changed message, or to send the message by exiting the <b>mail</b> program.
<code>~w filename</code>	Writes the message to the named file.
<code>~!cmd</code>	Pipes the message through the command <i>cmd</i> as a filter. If <i>cmd</i> gives no output or terminates abnormally, it retains the original text of the message. Otherwise, the output of <i>cmd</i> replaces the current message. The command <b>fmt</b> is often used as <i>command</i> to rejustify the message.
<code>~~</code>	Allows you to use the character <code>~</code> (tilde) in a message without it being interpreted as a command prefix. The sequence <code>~~</code> results in only one <code>~</code> being sent in the message. If you have changed the escape character, double that character instead of <code>~</code> to use the new escape character as a single character.

Figure 4 (Part 2 of 2). Mail Editor Commands

You can end a **mail** session with the **quit (q)** command. Messages that you have looked at go to your personal mailbox. Messages that you have marked to be deleted are deleted. Messages that you have not looked at go back to your system mailbox.



## Customizing the Mail Program

The **mail** command has a number of options that you can set to customize the mail system for your particular use. Use the **set** command to enable options, and the **unset** command to disable options. You can also use the **set** command to assign a value to an option.

The format for using the **set** command to enable options is:

```
set [option-list]
```

The *option-list* may be one or more options that you want to enable. To set options so that they are valid each time you use **mail**, put the commands in **.mailrc** in your \$HOME directory. In your \$HOME directory. To set options so that they are valid for all users on the system, put the commands in **/usr/lib/Mail.rc**. The following table, Figure 5, lists the binary options (those that need only be set or unset).

Option	Function
<b>append</b>	Causes messages saved in <b>mbox</b> to be appended (added to the end) rather than prepended (added to the beginning).
<b>ask</b>	Causes <b>mail</b> to prompt you for the subject of each message you send. If you respond with a new line (carriage return), no subject field is set.
<b>askcc</b>	Causes you to be prompted for the addresses of people to receive copies of the message. Responding with a new line indicates your satisfaction with the current list.
<b>autoprint</b>	Causes the <b>delete</b> command to behave like <b>dp</b> . Thus, after deleting a message, the next one is typed automatically.
<b>debug</b>	Same as specifying <b>-d</b> on the command line. Causes <b>mail</b> to display debugging information. <b>mail</b> does not send mail while in debug mode.
<b>dot</b>	Causes <b>mail</b> to interpret a period alone on a line as the terminator of a message you are sending.
<b>hold</b>	Holds messages in the system mailbox by default.
<b>ignore</b>	Causes interrupt signals from your terminal to be ignored and echoed as <b>@</b> 's.

Figure 5 (Part 1 of 2). Binary Options

## mail, Mail

---

Option	Function
<b>ignoreeof</b>	Related to <b>dot</b> . Makes <b>mail</b> refuse to accept an <b>Ctrl-D</b> as the end of a message. <b>ignoreeof</b> also applies to <b>mail</b> command mode.
<b>metoo</b>	Usually, when an alias containing the sender is expanded, the sender is removed from the expansion. Setting this option causes the sender to be included in the alias expansion (and thus receives copies of messages).
<b>nosave</b>	Normally, when a message is terminated with two interrupt sequences ( <b>Alt-Pause</b> ), <b>mail</b> copies the partial letter to the file <b>dead.letter</b> in your home directory. Setting the binary option <b>nosave</b> prevents this.
<b>Replyall</b>	Reverses the sense of the <b>reply</b> and <b>Reply</b> mailbox commands.
<b>quiet</b>	Suppresses the printing of the program banner when <b>mail</b> starts.
<b>verbose</b>	Same as using the <b>-v</b> flag on the command line. When <b>mail</b> runs in verbose mode, the actual delivery of messages is displayed on the user's terminal.

Figure 5 (Part 2 of 2). Binary Options

The following table, Figure 6 lists the valued options (those that need to be assigned a value).

Option	Function
<b>EDITOR</b>	Path name of the text editor to use in the <b>edit</b> command and <b>~e</b> escape. If not defined, then a default editor ( <b>/usr/bin/e</b> ) is used.
<b>PAGER</b>	Path name of the paging program to use for the <b>more</b> command or when the <b>crt</b> variable is set. If you do not specify a value for <b>PAGER</b> , the system uses <b>/bin/pg</b> .
<b>SHELL</b>	Path name of the shell to use in the <b>!</b> command and the <b>~!</b> escape. A default shell is used if this option is not defined.
<b>VISUAL</b>	Path name of the text editor to use in the <b>visual</b> command and <b>~v</b> escape. The default path name is <b>/usr/bin/vi</b> .

Figure 6 (Part 1 of 2). Valued Options



Option	Function
<b>crt = <i>n</i></b>	Calls the <b>pg</b> command to display the message when the message exceeds <i>n</i> lines.
<b>escape</b>	If defined, the first character of this option gives the character to use in the place of ~ to denote escapes.
<b>folder</b>	Defines the name of the directory to use for storing folders of messages. If this name begins with a /, <b>mail</b> considers it to be an absolute path name; otherwise, the folder directory is found relative to your home directory.
<b>record</b>	If defined, gives the path name of the file (relative to \$HOME) used to record all outgoing mail. If not defined, then outgoing mail is not saved. Do not include the home directory as part of the path name.
<b>screen</b>	If defined, controls the size of the window for message headers. You can set this option to show the number of lines on the screen. For example, the entry <b>screen=22</b> causes the system to scroll for 22 lines and then pause.
<b>toplines</b>	If defined, gives the number of lines of a message to be printed out with the <b>top</b> command; normally, the first five lines are printed.

Figure 6 (Part 2 of 2). Valued Options

## Flags

<b>-v</b>	Puts <b>mail</b> into verbose mode. Details of delivery are displayed on the user's terminal.
<b>-i</b>	Causes tty interrupt signals to be ignored. Useful when using <b>mail</b> on noisy phone lines.
<b>-n</b>	Inhibits the reading the <b>/usr/lib/Mail.rc</b> .
<b>-N</b>	Suppresses the initial printing of headers.
<b>-s <i>subject</i></b>	Specifies a subject for a message to be created.
<b>-f <i>name</i></b>	Causes <b>mail</b> to read in the contents of your <b>mbox</b> or the specified file for processing. When you <b>quit</b> , <b>mail</b> writes undeleted messages back to this file.
<b>-u <i>user-id</i></b>	Short way of doing <b>mail -f /usr/mail/<i>user-id</i></b> . Activates <b>mail</b> for a specified user's mailbox. You must have access permission to the specified mailbox.

## mail, Mail

---

### Files

/usr/mail/*	System mailboxes for all users.
\$HOME/mbox	Your personal mailbox.
\$HOME/.mailrc	File containing mail commands to customize <b>mail</b> to a specific user.
/tmp/R#	Temporary for editor escape.
/usr/lib/Mail.help	Help file for mailbox commands.
/usr/lib/Mail.tildehelp	Help file for mail editor commands.
/usr/lib/Mail.rc	File containing mail commands to change <b>mail</b> for all users on the system.

### Related Information

The following commands: “**bellmail**” on page 104, “**sendmail**” on page 897, and “**uucp**” on page 1144.

The chapter about sending and receiving mail in *IBM RT Using the AIX Operating System*.

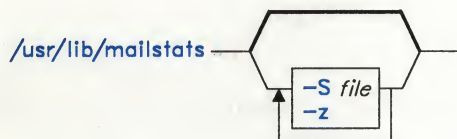
The chapters about Mail Handler (an alternative to **mail**) and about managing the mail system in *IBM RT Managing the AIX Operating System*.

## mailstats

### Purpose

Displays statistics regarding mail traffic.

### Syntax



AJ2FL146

### Description

This command reads the information in `/usr/adm/sendmail/sendmail.st` (or in the file specified with the `-S` flag), formats it, and writes it to standard output. The format of the information is shown in the following example:

```

Sendmail statistics from file "/usr/adm/sendmail/sendmail.st"
Collection started at Thu Feb 18 17:40:41 1988
  
```

Mailer	msgs_from	bytes_from	msgs_to	bytes_to
-----	-----	-----	-----	-----
local	1	2	1	201
prog	0	0	0	0
uucp	0	0	0	0
tcp	0	0	0	0

The fields in the report have the following meanings:

**Mailer** This field contains the name of the mailer program that handled the mail.

**msgs\_from**

This field (*messages from*) contains the number of messages that originated from the indicated mailer.

**bytes\_from**

This field contains the number of bytes of information in the messages sent from the indicated mailer.



## mailstats

---

**msgs\_to** This field (*messages to*) contains the number of messages that ended locally and were received by the indicated mailer.

**bytes\_to** This field contains the number of bytes of information in the messages received by the indicated mailer.

The collection start time indicated on the second line of the report is the time at which the first update to the empty file was performed.

If **sendmail** transmits mail directly to a file, such as **dead.letter** or an alias target, the message and byte counts are credited to the prog mailer in addition to the normal statistics for use of the prog mailer.

## Statistics Messages

When **mailstats** is called with no program flags, it can generate the following messages:

No statistics data in file `"/usr/adm/sendmail/sendmail.st"`

The **sendmail** program has not written any data into the statistics file.

mailstats: file size change; use previous mailstats version

The statistics file format is not the format expected by **mailstats**. Try using a previous version of **mailstats** to read it.

## Flags

**-S file** Specifies to use *file* as the input statistics file instead of `/usr/adm/sendmail/sendmail.st`

**-z** Clears the contents of the statistics file. Clearing the file erases the contents and allows you to start gathering statistics again.

## Files

`/usr/lib/mailstats`

The **mailstats** program.

`/usr/adm/sendmail/sendmail.st`

The database file containing mail system statistics.

`/usr/adm/sendmail/sendmail.cf`

The configuration file for **sendmail** program.

## Related Information

The command: "**sendmail**" on page 897.

The chapter about managing the mail system in *IBM RT Managing the AIX Operating System*.

---

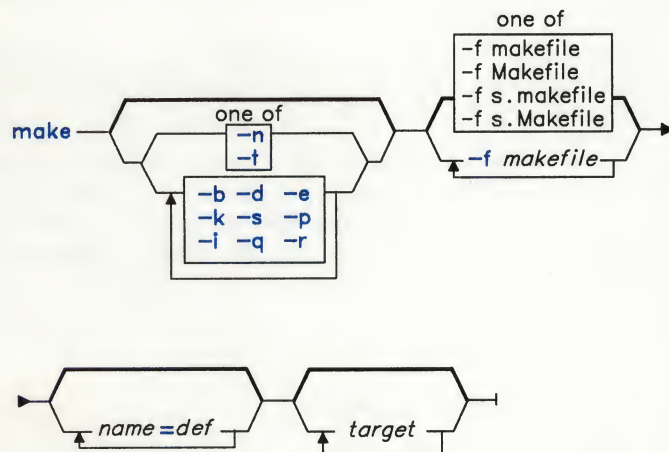
# make

---

## Purpose

Maintains up-to-date versions of programs.

## Syntax



OL805035

## Description

The **make** command reads *makefile* for information about the specified *target* files and for the commands necessary to update them. **make** does not change the *target* if you have not changed any of the source files since you last built it. It considers a missing file to be a changed file (out-of-date).

You can also include macro definitions on the command line after all of the flags. Macro definitions have the form:

*macro-name* = *string*

See "Macros" on page 628 for more information about macros and their uses.

The **make** command considers all entries on the command line that follow the flags and that do not contain an equal sign to be target file names.



## Description File

The description file contains a sequence of entries specifying the files that the target files depend on. The general form of an entry is:

```
targ [targ] . . . :[:][file] . . . [; cmd] . . . [#]
                                [cmd] . . . [#]
```

The first line of an entry (called the **dependency line**), contains a list of targets followed by a : (colon) and an optional list of prerequisite files or dependencies. If you put shell commands on the dependency line, they must be preceded by a ; (semicolon). All commands that follow the semicolon and all following lines that begin with a tab contain shell commands that **make** uses to build the target.

To specify more than one set of commands, you must enter more than one dependency definition. In this case, each definition must have the target name followed by two colons (::), a dependency list, and a command list.

The first line that does not begin with a tab or # (pound sign) begins a new dependency or a macro definition. Command lines are performed one at time, each by its own subshell. Thus, the effect of some shell commands, such as **cd**, does not extend across new-line characters. You can, however, put a \ (backslash) at the end of a line to continue it on the next physical line. A comment begins with a # and ends with a new-line character.

The first one or two characters in a command can be one of the following special characters:

- Ignores errors returned by the command on this line.
- @: Does not display this command line.
- @
- @- Does not display this command line and ignores errors.

## Suffixes

The **make** command has default rules that govern the building of most standard files. These rules depend on the standard suffixes used by the system utility programs to identify file types. These rules define the starting and ending file types so that, for example, given a specified **.o** file, **make** can infer the existence of a corresponding **.c** file and knows to compile it using the **CC -c** command.

A rule with only one suffix (that is, **.c:**) defines the building of *prog* from all its source files. Use a ~ (tilde) in the suffix to indicate a SCCS file. For example, the **.c~.o** rule governs changing an SCCS C source file into an object file. You can define rules within the description file. **make** recognizes as a rule any target that contains no slashes and starts with a dot.



You can also add suffixes to the list of suffixes recognized by **make** and add to the default dependency rules. Use the target name **.SUFFIXES** followed by the suffixes you want to add. Be careful of the order in which you list the suffixes. **make** uses the first possible name for which both a file and a rule exist. The default list is:

```
.SUFFIXES: .o .c .c~ .y
.y~ .l .l~ .s .s~
.sh .sh~ .h .h~
```

You can clear the list of suffixes by including **.SUFFIXES:** with no following list.

## Special Target Names

You can use some special target names in the description file to tell **make** to process the file in a different manner. The special target names are:

- .DEFAULT**      The commands that appear after this name in the description file tell **make** what to do if it can find no commands or default rules to tell it how to create a specific file.
- .IGNORE**        If this name appears on a line by itself, **make** does not stop when errors occur. Using a - (minus) as the first character on a line in the description file tells **make** to ignore errors for the command on that line.
- .PRECIOUS**      The files named on the same line as this special name are not removed when **make** is interrupted.
- .SILENT**        If this name appears on a line by itself, **make** does not display any of the commands that it performs to build a file.
- .SUFFIXES**      Use this name to add more suffixes to the list of file suffixes that **make** recognizes.

## Environment

When you run **make**, it reads the environment and treats all variables as macro definitions. **make** processes the environment variables after it processes its own internal rules and before processing any description files. Therefore, macro assignments in a description file normally override duplicate environment variables. The **-e** flag instructs **make** to use the environment variables instead of the description file macro assignments.

The **make** command recognizes a macro **MAKEFLAGS**, which can be assigned any **make** command line flag except **-f**, **-p**, and **-d**. When **make** begins, it assigns the current flags to **MAKEFLAGS**. It passes this variable to any commands it invokes, including additional invocations of **make** itself. Thus you can perform a **make -n** recursively on a software system to see what would have been performed. The **-n** is put in **MAKEFLAGS** and passed to further copies of the shell that runs the next level of **make** commands. In this way, you can check all of the description files for a software project without actually compiling the project.

**Note:** Some **makefile** macros can conflict with **cs**h variable substitutions. You should avoid using **make** with the **cs**h shell. The **sh** shell does not conflict with **make** macros and it is the recommended shell. Otherwise, you can avoid conflicts by adding a `SHELL = /bin/sh` to the makefile.

### Macros

Entries of the form `string1 = string2` are macro definitions. `string2` can consist of all characters that can occur on a line before a comment character (`#`) or before a new-line character that is not a continuation line. After this macro definition, **make** replaces each `$(string1)` in the file with `string2`. You do not have to use the parentheses around the macro name if the macro name is only one character long and there is no substitute sequence (see the next paragraph). If you use the following form, you can also replace characters in the macro string with other characters for one time that you use the macro:

```
$(string1[:subst1=[subst2]])
```

The optional `:subst1 = subst2` + specifies a substitute sequence. If you specify a substitute sequence, **make** replaces each `subst1` in the named macro with `subst2` (if `subst1` does not overlap with another `subst1`). Strings in a substitute sequence begin and end with any of the following: a blank, tab, new-line character, or beginning of line. See “Libraries” on page 629 for an example of the use of the substitute sequence.

**Note:** Because **make** uses the dollar sign symbol (`$`) to designate a macro, do not use that symbol in file names of targets and parents, or in commands in the description file unless you are using a defined **make** macro.

### Internal Macros

The **make** command has five internal macros. It assigns values to these macros under one or more of the following conditions:

- When it uses an internal rule to build a file.
- When it uses a **.DEFAULT** rule to build a file.
- When it uses rules in the description file to build a file.
- When the file is a library member.

They are defined as follows:

- `$*` The file name (without the suffix) of the source file.
- `$@` The full target name of the current target.
- `$<` The source files of an out-of-date module. **make** evaluates this macro when applying inference rules or the **.DEFAULT** rule. For example:

```
.C.O:  
cc -c $<
```



Here, `$<` is the equivalent of `$*` and refers to the `.c` file of any out-of-date `.o` file.

`$?` The list of out-of-date files. **make** evaluates this macro when it evaluates explicit rules from *makefile*.

`$$` The name of an archive library member. **make** evaluates this macro only if the target is an archive library member of the form *lib(file.o)*. In this case, `$@` evaluates to *lib* and `$$` evaluates to the library member, *file.o*.

You can add an uppercase **D** or **F** to indicate “directory part” or “file part,” respectively, to all internal macros except for `$?`. Thus, `$(@D)` refers to the directory part of the name `$@`. If there is no directory part, **make** uses `./`.

## Libraries

If a target name contains parentheses, **make** considers it an archive library. The string within parentheses refers to a library member. Thus, *lib(file.o)* and `$(LIB)(file.o)` both see an archive library which contains *file.o*. (You must have already defined the **LIB** macro.) The expression `$(LIB)(file1.o file2.o)` is not legal.

Rules that apply to archive libraries have the form *x.a*, where *x* is the suffix of the file you want to add to an archive library. For example, *.c.a* indicates a rule that changes any C source file to a library file member. The following lines give the default rule for this change:

```
lib: lib(file1.o) lib(file2.o) lib(file3.o)
    @echo lib is now up to date
.c.a:
    $(CC) -c $(CFLAGS) $<
    ar rv $@ $*.o
    rm -f $*.o
```

*x* must be different from the suffix of the archive member. Therefore, you cannot have *lib(file.o)* depend upon *file.o*.

Another, but more limited, example of an archive library maintenance rule follows:

```
lib: lib(file1.o) lib(file2.o) lib(file3.o)
    $(CC) -c $(CFLAGS) $(?:.o=.c)
    ar rv lib $?
    rm $? @echo lib is now up to date
.c.a;;
```

This example rule uses a substitute sequence (`.o=.c`) to replace with `.c` files all `.o` files generated by the `$?` macro. The `$?` list is the set of object file names (inside *lib*) with C source files that are out of date. The macro substitution translates `.o` to `.c`.



## make

---

If this rule appears in your description file, it disables the default `.c.a:` rule, which creates each object file one by one. This type of organization speeds up archive library maintenance, but becomes hard to use if the archive library contains a mix of assembly programs and C programs.

## Flags

- b** Recognizes *makefiles* that were written for old versions of **make**.
- d** Displays detailed information about the files and times that **make** examines (debug mode).
- e** Uses environment variables in place of any assignments made within description files. These assignments normally replace environment variables.
- f *makefile*** Reads *makefile* for a description of how to build the target file. If you give only a - (minus) for *makefile*, **make** reads standard input. If you do not use the **-f** flag, **make** looks in the current directory for a description file named **makefile**, **Makefile**, **s.makefile**, or **s.Makefile**. You can specify more than one description file by entering the **-f** flag more than once (with its associated *makefile* parameter).
- i** Ignores error codes returned by commands. **make** normally stops if a command returns a nonzero code. Use this flag to compile several modules only if you want **make** to continue when an error occurs in one of the modules. Do not link the resulting modules when you use this flag.
- k** Stops processing the current target if an error occurs, but continues with other branches that do not depend on that target.
- n** Displays commands, but do not run them. Displays lines beginning with an @ (at sign). If the command in the description file contains the string `$(MAKE)`, perform another call to **make** (see the discussion of the **MAKEFLAGS** macro on page 627). Use this flag to preview the performance of **make**.
- p** Displays the complete set of macro definitions and target descriptions before performing any commands.
- q** Returns a zero status code if the target file is up to date; returns a nonzero status code if the target file is not up to date.
- r** Does not use the default rules.
- s** Does not display commands on the screen as they are performed.

- t** Changes only the date of the files, rather than performing the listed commands. Use this flag if you have made only minor changes to a source file that do not affect anything outside of that file. This flag changes the date of all target files that appear on the command line or in the description file.

## Examples

1. To make the file specified by the first entry in the description file:

```
make
```

2. To display, but not run, the commands that **make** would use to make a file:

```
make -n search.o
```

You may want to do this to verify that a new description file is correct before using it.

3. To save the internal rules in a file:

```
make -p -f /dev/null 2> /dev/null > defaults
```

This lists the internal rules and macros and saves them in the file `defaults` for viewing or editing. All exported shell environment variables are included in the list of macro definitions.

## Files

Makefile  
makefile  
s.Makefile  
s.makefile

## Related Information

The discussion of **make** in *AIX Operating System Programming Tools and Interfaces*.

# makedbm

---

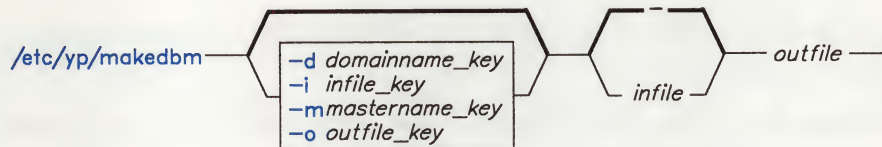
## makedbm

---

### Purpose

Makes a Yellow Pages **dbm** map.

### Syntax



A5AC5004

### Description

The **makedbm** command is most often invoked from **/etc/yp/Makefile** to generate Yellow Pages maps. The **makedbm** command converts *infile* to a pair of files in **dbm** format. The two files are **outfile.pag** and **outfile.dir**. Each line in the input file is converted to a single **dbm** record. All characters up to the first space or tab form the key while the rest of the line is the value data. If a line ends with \ (backslash), data for that record is continued on the next line. Yellow Pages clients must interpret the # symbol since **makedbm** does not treat it as a comment character. If *infile* is - (minus sign), **makedbm** reads standard input.

The **makedbm** command generates a special entry with the key **YP-LAST-MODIFIED** giving the date of *infile* or the current time if *infile* is specified to be standard input.

---

#### Japanese Language Support Information

If Japanese Language Support is installed on your system, this command is not available.

---

### Flags

- d Creates a special entry with the key **YP-DOMAIN-NAME**.
- i Creates a special entry with the key **YP-INPUT-FILE**.
- o Creates a special entry with the key **YP-OUTPUT-NAME**.



- m**      Creates a special entry with the key *YP-MASTER-NAME*. If no master host name is specified, *YP-MASTER-NAME* is set to the local host name.
- u**      Displays individual entries in a **dbm** file, with a single space separating keys from values.

## Example

In the following example, data from the */etc/passwd* file is converted to a form that **makedbm** can use to create the Yellow Pages map **passwd.nam**:

```
awk '{FS=":"; OFS="\t"; print $1,$0}' \  
etc/passwd | makedbm - passwd.nam
```

## File

*/etc/yp/Makefile*

## Related Information

The following commands: “**yppush**” on page 1252 and “**ypinit**” on page 1243.

## makekey

---

## makekey

---

### Purpose

Generates an encryption key.

### Syntax

`/usr/lib/makekey` —

OL805240

### Description

The **makekey** command generates an encryption key to use with programs that perform encryption. Its input and output are usually pipes.

The **makekey** command reads 10 characters from standard input and writes 13 characters to standard output. The first 8 of the 10 input characters can be any sequence of ASCII characters. The last two input characters (the *salt*), are best chosen from the set [a-zA-Z0-9.,/]. The salt characters are repeated as the first two characters of the output. The remaining 11 output characters are chosen from the same set as the salt and constitute the output key that you use as the *key* parameter to programs that perform encryption.

---

#### Japanese Language Support Information

This command has not been modified to support Japanese characters.

---

### Example

To generate an encryption key:

```
/usr/lib/makekey  
1234567890
```

This generates an encryption key based on the string 1234567890. The key 90y74T/NXw1U is displayed at the work station. Do *not* press **Ctrl-D** after typing the input key 1234567890 because this would end your shell session. Also, the shell prompt appears immediately after the generated key, instead of appearing on a separate line as it usually does. This is normal.

---

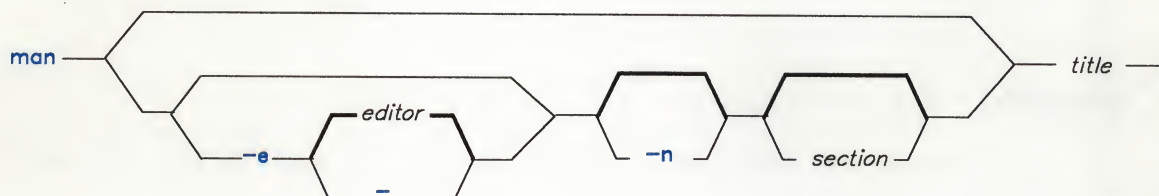
## man

---

### Purpose

Displays manual entries online.

### Syntax



AJ2FL103

### Description

The **man** command locates and displays the entries for the specified *title* and *section* of the online manual. If specified, **man** displays entries through *editor*. If an *editor* is not specified, **man** displays entries by default through the editor set by the **EDITOR** environment variable. If the **EDITOR** environment variable is not set, then the entry is displayed on standard output.

The section number may be one of the following:

- Commands and Application Programs
- System Calls
- Subroutines
- Special Files
- File Formats
- Games
- Miscellaneous Facilities.

If no section is specified, then all sections are searched for each title and all such occurrences are displayed; if a given title is not found in the specified section, then all sections are searched for that title.

**Note:** This command is available optionally; it is not part of the standard AIX Operating System package.



# man

---

## Flag

- `-e editor` Displays entries with the specified *editor*.
- `-e -` Does not display entries with the *editor*.
- `-n` Displays entries without pagination.

## Files

```
/usr/bin/man  
/usr/man/cat[1-7]/*  
/usr/bin/man[i l n o p]
```

## Examples

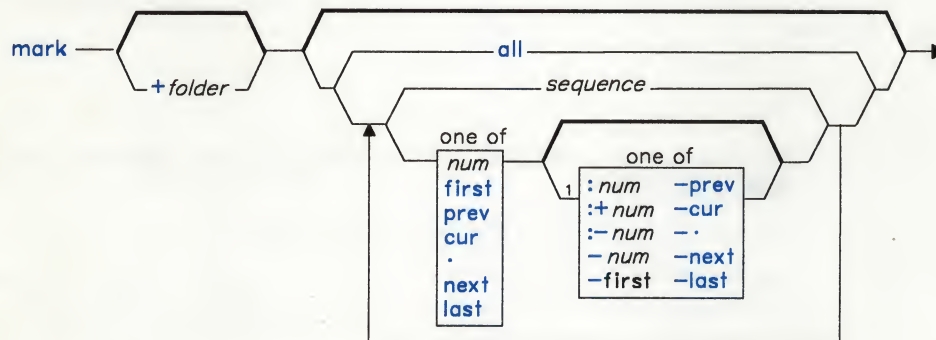
1. To display the **man** entry on standard output when the **EDITOR** variable is not set:  
`man man`
2. To view section 1 of this entry in the **vi** editor:  
`man -e vi 1 man`

# mark

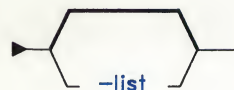
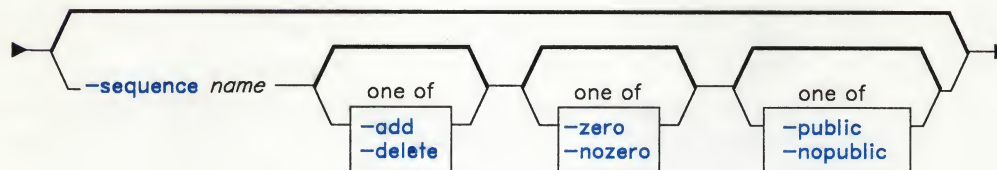
## Purpose

Creates, modifies, and displays message sequences.

## Syntax



AJ2FL200



mark — -help —

AJ2FL201

<sup>1</sup> Do not put a blank between these items.

OL805308

## Description

The **mark** command is used to create a sequence, delete a sequence, add messages to a sequence, and delete messages from a sequence. The **mark** command is also used to list messages in a sequence and list sequences in a folder. The **mark** command is part of the MH (Message Handling) package and can be used with other MH and AIX commands.

The **mark** command specified with only a folder name lists the sequences defined for that folder and the messages that comprise each of the sequences. If you specify a new sequence name with the **-sequence** flag, **mark** creates a new sequence. You can use the **-add** flag to add messages to a sequence and the **-delete** flag to remove messages from a sequence. When all messages are deleted from a sequence, **mark** removes the sequence name from the folder.

## Flags

- |                     |   |                 |              |             |            |          |             |             |            |                 |
|---------------------|---|-----------------|--------------|-------------|------------|----------|-------------|-------------|------------|-----------------|
| <b>-add</b>         | Adds messages to the specified sequence. If messages are specified, <b>-add</b> is the default.   |                 |              |             |            |          |             |             |            |                 |
| <b>-delete</b>      | Deletes messages from the specified sequence.   |                 |              |             |            |          |             |             |            |                 |
| <b>+folder msgs</b> | Specifies messages that you want <b>mark</b> to select from. <i>msgs</i> can be several messages, a range of messages, or a single message. You can use the following message references when specifying <i>msgs</i> :<br><table border="0" data-bbox="525 867 1087 954"><tr><td><i>num</i></td><td><b>first</b></td><td><b>prev</b></td></tr><tr><td><b>cur</b></td><td><b>.</b></td><td><b>next</b></td></tr><tr><td><b>last</b></td><td><b>all</b></td><td><i>sequence</i></td></tr></table> <p>If the <b>-list</b> flag is used, the default for <i>msgs</i> is <b>all</b>. Otherwise, the default is the current message. The default folder is the current folder. If you specify a folder, that folder becomes the current folder.</p> | <i>num</i>      | <b>first</b> | <b>prev</b> | <b>cur</b> | <b>.</b> | <b>next</b> | <b>last</b> | <b>all</b> | <i>sequence</i> |
| <i>num</i>          | <b>first</b>  | <b>prev</b>     |              |             |            |          |             |             |            |                 |
| <b>cur</b>          | <b>.</b>  | <b>next</b>     |              |             |            |          |             |             |            |                 |
| <b>last</b>         | <b>all</b>  | <i>sequence</i> |              |             |            |          |             |             |            |                 |
| <b>-help</b>        | Displays help information for the command.  |                 |              |             |            |          |             |             |            |                 |
| <b>-list</b>        | Displays the messages in the specified sequence. If you do not specify a sequence, <b>-list</b> displays all sequence names defined for the folder and the messages in each sequence.   |                 |              |             |            |          |             |             |            |                 |
| <b>-npublic</b>     | Restricts the specified sequence to your usage. <b>-npublic</b> does not restrict the messages in the sequence, only the sequence. This option is the default if the folder is write-protected from other users.  |                 |              |             |            |          |             |             |            |                 |
| <b>-nozero</b>      | Modifies the sequence by adding or deleting only the specified messages (see the <b>-zero</b> flag). This flag is the default.  |                 |              |             |            |          |             |             |            |                 |
| <b>-public</b>      | Makes the specified sequence available to other users. <b>-public</b> does not make protected messages available, only the sequence. This flag is the default if the folder is not write-protected from other users.  |                 |              |             |            |          |             |             |            |                 |



- sequence name** Specifies a sequence for the **-list**, **-add**, and **-delete** operations. You must specify this flag for the **-add** and **-delete** operations.
- zero** Clears the specified sequence of all messages before adding any other messages. When the **-delete** flag is also specified, **-zero** places all of the messages from the folder into the sequence before deleting any messages.

## Profile Entries

- Current-Folder:** Sets your default current folder.
- Path:** Specifies your *user-mh-directory*.

## Files

`$HOME/.mh-profile` The MH user profile.

## Related Information

The MH command “**pick**” on page 748.

The **mh-profile** file in *AIX Operating System Technical Reference*.

“Overview of the Message Handling Package” in *Managing the AIX Operating System*.

## mdrc

---

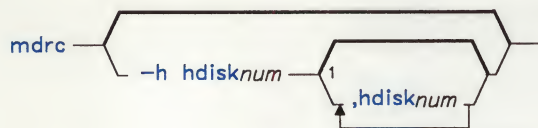
## mdrc

---

### Purpose

Allows you to reinstall a user-created minidisk after you have reinstalled AIX.

### Syntax



OL805440

### Description

The **mdrc** command provides access to user-created minidisks. You should run this command if you have reinstalled the AIX Operating System or if you have had to replace the **/etc/system**, **/etc/filesystems**, or **/etc/ddi/cpmgr** files with copies that do not contain stanzas describing any user-installed minidisks. The system uses the information in these stanzas to configure the minidisks at system startup, and **mdrc** recreates the necessary stanzas. Normally, **mdrc** uses the backup copy of **/etc/filesystems** produced by the **minidisks** command when you use it to create a new minidisk. This backup copy is named **/u/filesystems**.

If **mdrc** cannot recreate the original **/etc/filesystems** stanza for AIX Operating System minidisks, it assigns attributes of **Auto Mount=no**, **Read/Write Status=R/W**, and **Mount Directory=/tmp/directory/hdn** to the minidisk. In this case, you should then run the **minidisk** command to change the attributes to the values you want. You might also need to run the **mkdir** command to create the mount directory, if you reinstalled the entire AIX Operating System.

If a minidisk has been created for use by the Personal Computer AT Coprocessor<sup>4</sup>, **mdrc** will update the **/etc/ddi/cpmgr** file. If you have not installed Personal Computer AT Coprocessor Services before running **mdrc**, it creates an entry in **/etc/system**, but displays a warning message because the **/etc/ddi/cpmgr** does not exist. You must run **mdrc** again after you install the Coprocessor to be able to use the coprocessor minidisks.

---

<sup>4</sup> Personal Computer AT Coprocessor is a registered trademark of International Business Machines Corporation.

The **mdrc** command does not recognize external disks, or any minidisks on them, if the disks are not configured. To configure an external disk and its minidisks, see “**varyon**” on page 1180.

You must have superuser authority or be a member of the system group to run the **mdrc** command. When auditing is on, an audit record of the type **mdrc** is created.

## Flag

**-h** *hdisknum*[, *hdisknum*] . . .

Specifies any disks that have been removed or damaged and tells **mdrc** to remove the minidisk configuration entries for these disks. If you do not specify this flag and an external disk is not configured, **mdrc** ignores entries in the configuration files for the external disk’s minidisks.

**Note:** If you do not have any external disks, you do not need to specify this flag.

## Files

/etc/filesystems  
/etc/system  
/etc/ddi/cpmgr

## Related Information

The following commands: “**watch**” on page 1209, “**mkdir**” on page 657, and “**varyon**” on page 1180.

The **filesystems** and **system** files in *AIX Operating System Technical Reference*.



## mesg

---

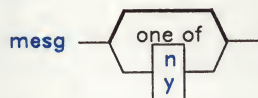
## mesg

---

### Purpose

Permits or refuses **write** messages.

### Syntax



OL805036

### Description

The **mesg** command controls whether other users on the system can send messages to you with the **write** command. Called without arguments, **mesg** displays the current work station message-permission setting. The shell startup process permits messages by default. You can override this default action by including the line: **mesg n** in your **\$HOME/.profile** file. A user with superuser authority can send **write** messages to any work station, regardless of its message permission setting. Message permission has no effect on messages delivered through the electronic mail system (**sendmail**).

### Flags

- n** Disables incoming **write** messages. Use this form of the command to avoid having others clutter your display with incoming messages.
- y** Permits **write** messages.

### Files

/dev/tty\*

### Related Information

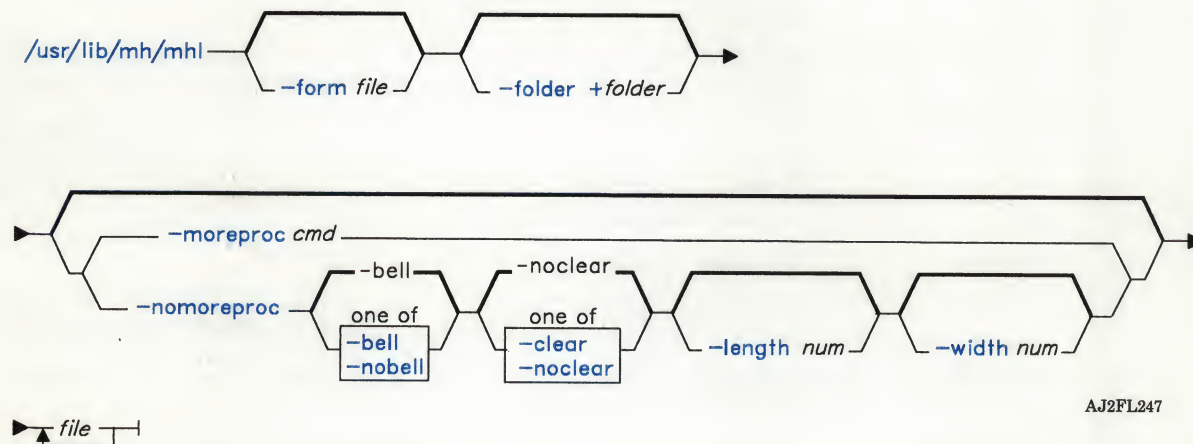
The following commands: “**write**” on page 1225 and “**sendmail**” on page 897.

# mhl

## Purpose

Produces formatted listings of messages.

## Syntax



AJ2FL247

```

/usr/lib/mh/mhl -help
  
```

AJ2FL248

## Description

The **mhl** command is used to create formatted lists of messages. The **mhl** command is part of the MH (Message Handling) package and can be used with other MH and AIX commands. **mhl** is usually invoked through the profile entry **showproc:** or through the **-showproc** flag in other MH commands.

The **mhl** command uses the formatting directions listed in the format file to display the message information about each of the specified messages. If you specify more than one message, **mhl** provides a prompt before displaying each screen of messages or with **-nomoreproc**, before displaying each message. If you specify **-nomoreproc**, press **Enter** or **END OF FILE** to see the next message. Press **INTERRUPT** to stop the current message output and to receive a prompt for the next message. Press **QUIT** to stop the command output.

## Flags

- bell** Produces an audible indicator at the end of each page. This flag affects **mhl** only if the output device is a display and the **moreproc:** entry is defined and empty. This flag is the default.
- clear** Clears the screen at the end of each page when the output device is a display. When the output device is not a display, **-clear** inserts a form feed character at the end of each message. If the output device is a display, **mhl** uses the **\$TERM** and **\$TERMCAP** environment variables to determine the type of display. This flag affects **mhl** only if the **moreproc:** entry is defined and is empty.
- folder +folder** Specifies the folder to be used for the **mhl.format messagename** entry. The default is the value of the **\$mhfolder** environment variable.
- form file** Uses the format contained in the specified file. If you do not specify this flag, **mhl** uses the format described in *user-mh-directory/mhl.format*. If this file does not exist, **mhl** uses the system default format described in */usr/lib/mh/mhl.format*.
- help** Displays help information for the command.
- length num** Sets the length of the output. The default value is the value indicated by **\$TERMCAP**. If that value is not appropriate, the default value is 40.
- moreproc cmd** Uses *cmd* instead of the value of the **moreproc:** entry specified in **\$HOME/.mh-profile**.
- nobell** Does not produce an audible indicator at the end of each page. This flag affects **mhl** only if the output device is a display and the **moreproc:** entry is defined and is empty.
- noclear** Does not clear the screen at the end of each page when the output device is a display. When the output device is not a display, **-clear** does not insert a form feed character at the end of each message. This flag affects **mhl** only if the **moreproc:** entry is defined and is empty. This flag is the default.
- nomoreproc** Sets **moreproc:** as an empty value.
- width num** Sets the width of the output. The default value is the value indicated by **\$TERMCAP**. If that value is not appropriate, the default value is 80.



## Profile Entry

**moreproc:** Specifies the interactive program for communicating with user.

## Files

<code>/usr/lib/mh/mhl.format</code>	The default MH message template.
<code>user-mh-directory/mhl.format</code>	The user's default message template. (If it exists, it overrides the default MH message template.)
<code>\$HOME/.mh-profile</code>	The MH user profile.

## Related Information

Other MH commands: “**ap**” on page 53, “**dp**” on page 352, “**next**” on page 694, “**prev**” on page 765, and “**show**” on page 942.

The **mh-format** and **mh-profile** files in *AIX Operating System Technical Reference*.

“Overview of the Message Handling Package” in *Managing the AIX Operating System*.

# mhmail

---

## mhmail

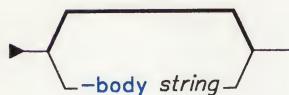
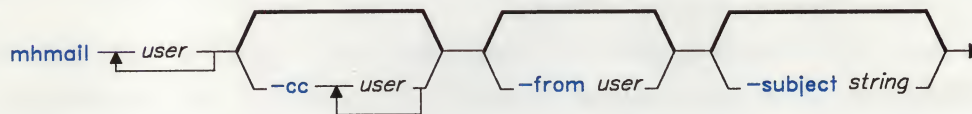
---

### Purpose

Sends or receives mail.

### Syntax

mhmail —|



mhmail — -help —|

AJ2FL236

### Description

The **mhmail** command is used to incorporate messages and compose messages. The **mhmail** command is part of the MH (Message Handling) package and can be used with MH and AIX commands.

The **mhmail** command entered by itself incorporates messages from your mailbox. If you specify user addresses, **mhmail** accepts text from your terminal and composes a message. You can end the message text by pressing **END OF FILE**. **mhmail** sends a copy of the message to each specified address.

## Flags

- body *string***      Sends a message with *string* as the body. When you specify **-body**, **mhmail** does not accept text from the terminal.
- cc *users***          Sends a copy of the message to the specified users. **mhmail** puts the addresses in the **cc:** field.
- from *user***          Places the specified user address in the **From:** field of the message.
- help**                  Displays help information for the command.
- subject *string***      Places the specified text string in the **Subject:** field of the message.

## Files

`/usr/mail/$USER`                  The location of the mail drop.

## Related Information

Other MH commands: “**inc**” on page 518 and “**post**” on page 758.

The **mh-alias**, **mh-mail**, and **mh-profile** files in *AIX Operating System Technical Reference*.

“Overview of the Message Handling Package” in *Managing the AIX Operating System*.



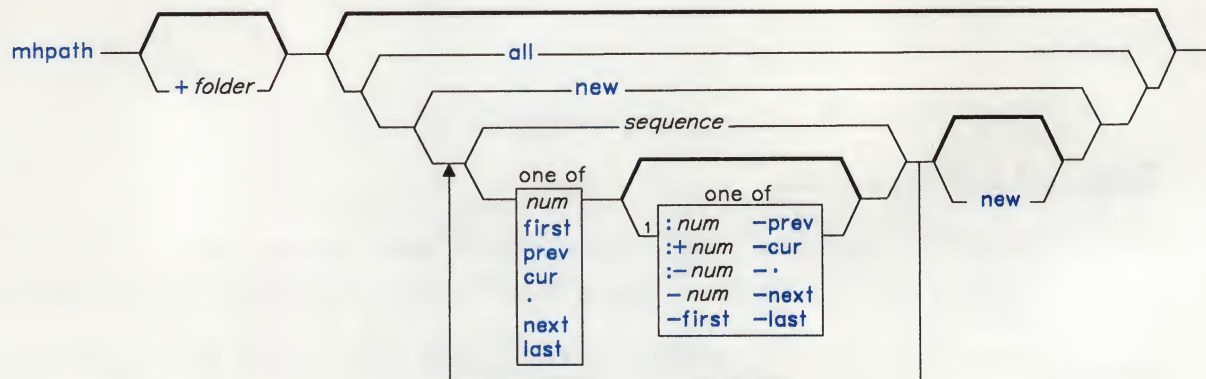
# mhpath

## mhpath

### Purpose

Prints full path names of messages and folders.

### Syntax



`mhpah` — `-help` —

<sup>1</sup> Do not put a blank between these items.

AJ2FL214

AJ2FL215

OL805308

### Description

The **mhpah** command is used to list the path names of folders and messages. **mhpah** is part of the MH (Message Handling) package and can be used with other MH and AIX commands.

The **mhpah** command lists the path names of all specified messages. If you do not specify any messages, **mhpah** lists the path name of the folder. If you do not specify messages or a folder, **mhpah** lists the path name of the current folder.

## Flags

**+folder msgs** Specifies the folder or the messages for which you want to list path names. *msgs* can be several messages, a range of messages, or a single message. You can use the following message references when specifying *msgs*:

<i>num</i>	<b>first</b>	<b>prev</b>	<b>cur</b>
.	<b>next</b>	<b>last</b>	<b>new</b>
<b>all</b>	<i>sequence</i>		

You cannot use **new** in a message range.

If you do not specify a message, **mhpath** lists the path name of the specified folder. The default folder is the current folder.

**-help** Displays help information for the command.

## Profile Entries

**Current-Folder:** Sets your default current folder.  
**Path:** Specifies your *user-mh-directory*.

## Files

*\$HOME/.mh-profile* The MH user profile.

## Related Information

The MH command “**folder**” on page 429.

The **mh-mail**, and **mh-profile** files in *AIX Operating System Technical Reference*.

“Overview of the Message Handling Package” in *Managing the AIX Operating System*.

## minidisks

---

## minidisks

---

### Purpose

Adds, deletes, changes, and displays minidisks.

### Syntax

`minidisks` —

OL805307

### Description

The **minidisks** command lets you add, delete, show, or change characteristics of a minidisk. To use the **minidisks** command, you must be a member of the system group or have superuser authority. When a minidisk is added or deleted, an audit record of the type **minidisk - add** or **minidisk - del** is created. When a minidisk is changed, an audit record of the type **stanza - add** and **stanza - del** is created.

The **minidisks** command is menu-driven. For information on how to use it, see *Installing and Customizing the AIX Operating System*.

### Files

/dev	Directory
/tmp	Directory
/etc/ddi	Directory
/etc/master	
/etc/system	
/etc/mdkaf	
/etc/filesystems	
/tmp/CONFIGREPORT	

### Related Information

The following command: “**mkdir**” on page 657.

The discussion of **minidisks** in *Installing and Customizing the AIX Operating System*.

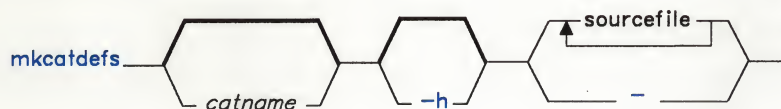


## mkcatdefs

### Purpose

Preprocesses a message source file.

### Syntax



OL805486

### Description

The **mkcatdefs** message facility program preprocesses a message source file containing symbolic identifiers<sup>5</sup>, allowing **gencat** to create the **symbname** message catalog. The format for **mkcatdefs** is:

```
$ mkcatdefs symbname sourcefile
```

The *sourcefile* message source file contains symbolic identifiers.

The **mkcatdefs** program produces the *symbname-msg.h* file containing definition statements equating your symbolic identifiers with set numbers and message ID numbers assigned by **mkcatdefs**. The *symbname-msg.h* file is required in your application program if you use symbolic identifiers.

The **mkcatdefs** program sends message source data, with numbers instead of symbolic identifiers, to standard output. This output is suitable as input to the **gencat** program. You can use the **>** (redirection symbol) to write the message source to a file, then use the file as input to **gencat**, or use the **runcat** shell script described in “**runcat**” on page 852. You can create a message text source file, using any text editor to enter the messages.

Assign message set numbers and message ID numbers to each message by using the commands described in this section.

<sup>5</sup> Symbolic references are not defined by X/Open, but are an AIX extension.

## mkctdefs

---

Use the **\$set** command in a source file to give a group of messages a set number. The format of the **\$set** command is:

```
$set n [comment]
```

The message number is specified by *n*. Instead of a number, you can specify a symbolic identifier, which must contain only letters, digits, or the **\_** (underscore character). The maximum length of an identifier is 65 characters. The **mkcatdefs** program assigns a set number to the identifier. The assigned set number is one higher than the preceding set number, or 1 if it is the first **\$set**.

Note that **mkcatdefs** inserts a **\$delset** before the **\$set** in the output message source file.

You can mix numbers and symbolic identifiers.

You can include a comment in the **\$set** command, but it is not required. The following example includes a comment:

```
$set CEM Communication Error Messages
```

Use the **\$delset** command to remove all of the messages belonging to the specified set from a catalog. The format of the **\$delset** command is:

```
$delset n [comment]
```

The message set is specified by *n*. The **\$delset** command must be placed in the proper set number order with respect to any **\$set** commands in the same source file. You can include a comment in the **\$delset** command also.

You can include a comment line anywhere in the source file, except within message text. Indicate comments as shown below:

```
$ [comment]
```

You must leave at least one space after the **\$**.

Enter the message text and symbolic message identifier as follows:

*ID message-text*

*ID* can be either a number or a symbolic identifier and can contain only letters, digits, or the **\_** (underscore character). The maximum length of an identifier is 65 characters. The **mkcatdefs** program assigns a message number to the identifier. The assigned number is one higher than the preceding message number, or 1 if it is the first message after the **\$set** command.

Note that **mkcatdefs** inserts a **\$delset** before the **\$set**, which means you cannot add, delete, or replace single messages in the catalog if you are using symbolic message identifiers. You must enter all messages in the set.

You can mix numbers and symbolic identifiers.



You must leave at least one space after the message identifier or number.<sup>6</sup> All text following the first nonblank character is included in the message text, to the end of the line. If the source contains a `$quote` command preceding the message, all text between the two quotation marks is included. Use the `\` (escape character) to continue message text on the following line. The `\` must be the last character on the line, as in the following example:

```
FIVE      Text associated with \
message FIVE.
```

These two lines define the single-line message:

```
FIVE      Text associated with message FIVE.
```

The `\` can be used to include special characters in the message text. These special characters are defined as follows:

- |                                |   |
|--------------------------------|---|
| <code>\n</code>                | Performs a new-line function when the message is displayed.   |
| <code>\t</code>                | Inserts a horizontal tab character when the message is displayed.   |
| <code>\v</code>                | Inserts a vertical tab when the message is displayed.   |
| <code>\b</code>                | Performs a backspace function when the message is displayed.  |
| <code>\r</code>                | Inserts a carriage-return character when the message is displayed.  |
| <code>\f</code>                | Inserts a form feed character when the message is displayed.  |
| <code>\\</code>                | Displays the <code>\</code> (backslash) character in the message.   |
| <code>\ddd</code>              | Displays the single-byte character associated with the octal value represented by the valid octal digits <i>ddd</i> . One, two, or three octal digits can be specified; however, you must include leading zeros if the characters following the octal digits are also valid octal digits. For example, the octal value for \$ is 44. To display \$5.00 use <code>\0445.00</code> , not <code>\445.00</code> , or the 5 will be parsed as part of the octal value. |
| <code>\xddd<sup>7</sup></code> | Displays the single-byte or double-byte character associated with the hexadecimal value represented by the four valid hexadecimal digits <i>ddd</i> . You can specify one, two, three, or four digits, but you must include leading zeros to avoid parsing errors (see <code>\ddd</code> ).   |

---

<sup>6</sup> AIX allows any amount of white space after the message ID number; however X/Open specifies that you leave only one space between the message number and the message text.

<sup>7</sup> This escape sequence is an AIX extension to X/Open.



You can also include **printf** conversion specifications in messages that are displayed by applications using **printf** or **NLprintf** (see **printf** in *AIX Operating System Technical Reference*). If you display a message from a shell script with **dspmsg**, the message can contain the **%s** or **%n\$s** conversion specifications (see “**dspmsg**” on page 359).

You can use the **\$quote** command in a message source file to define a character for delimiting message text. The format for this command is:

```
$quote [char] [comment]
```

Use the specified character before and after the message text as shown in the following example source file:

```
$quote "      Use a double quotation mark to delimit message text

$set MSFAC          Message Facility - symbolic identifiers

SYM_FORM   "Symbolic identifiers can only contain alphanumeric \
characters or the _ (underscore character)\n"

SYM_LEN    "Symbolic identifiers cannot be more than 65 \
characters longn \"

5           "You can mix symbolic identifiers and numbers \n"

$quote

MSG_H      Remember to include the "msg_h" file in your program\n
```

In this example, the **\$quote** command sets the quote character to " then disables it before the last message, which contains quotation marks.

When you process this file with **mkcatdefs**, the modified source is written to standard output. Standard output can either be redirected to a file using the **>** (redirection symbol) or piped to **gencat** see “**gencat**” on page 470.

The following source is created:

\$quote "      Use double quotation marks to delimit message text

\$delset 1  
\$set 1

1      "Symbolic identifiers can only contain alphanumeric \ characters or the \_ (underscore character)\n"

2      "Symbolic identifiers cannot be more than 65 \ characters long\n"

5      "You can mix symbolic identifiers and numbers\n"

\$quote

6      remember to include the "msg-h" file in your program

Note that the assigned message numbers are noncontiguous because the source contained a specific number. The **mkcatdefs** program always assigns the previous number plus 1 to a symbolic identifier.

The **mkcatdefs** program also produces a definition file for inclusion in your program. The name of the file is *symname*, entered as the first parameter to the **mkcatdefs** command. (If you specify the *-h* flag instead of the *symname*, no definition file is produced.)

If the symbolic source defined above were in a file called *symb.src*, you could use the **mkcatdefs** command as follows:

```
$ mkcatdefs symb symb.src >symb.msg
```

The generated *symb-msg.h* file would look as follows:

```
#include <limits.h>
#include <nl_types.h>
#define MF_SYMB "symb.cat"
```

```
/* The following was generated from symb.src. */
```

```
/* definitions for set MSFAC */
#define MSFAC 1

#define SYM_FORM 1
#define SYM_LEN 2
#define MSG_H 6
```

## mkctdefs

---

Note that **mkcatdefs** also created a symbol **MF\_SYMB** by adding **MF\_** to the *symbname*, in uppercase letters. The **mkcatdefs** program assumes that the name of the generated catalog should be *symbname.cat*, and generates this symbol for your use with **catopen** or **NLcatopen**.

Since this file includes **limits.h** and **nl\_types.h**, you do not need to include them in your application program. (**nl\_types** defines special data types required by the message facility routines.)

---

### Japanese Language Support Information

If Japanese Language Support is installed on your system, this command is not available.

---

## Flags

**-h**           Supresses the generation of a **...msg.h** file.

## Related Information

The following commands: “**dspcat**” on page 357, “**dspmsg**” on page 359, “**gencat**” on page 470, and “**runcat**” on page 852.

The **catgets**, **catgetamsg**, **catclose**, **NLcatopen**, **NLcatgets**, and **NLgetamsg** files in *AIX Operating System Technical Reference*.

The discussion of **mkcatdefs** in *AIX Operating System Programming Tools and Interfaces*.



---

# mkdir

---

## Purpose

Makes a directory.

## Syntax

`mkdir` 

OL805037

## Description

The **mkdir** command makes a new *directory* in either the local or a remote node. **mkdir** creates the new directories with read, write, and execute permissions enabled for all users. You can change the permissions it sets by default with the **umask** command (see page 1110). **mkdir** also creates by default the standard entries `.` (dot), for the directory itself, and `..` (dot dot), for its parent.

**Note:** To make a new *directory* you must have write permission in the parent directory.

## Related Information

The following commands: “**sh**” on page 913, “**rm**” on page 833, and “**umask**” on page 1110.

The **mkdir** system call in *AIX Operating System Technical Reference*.

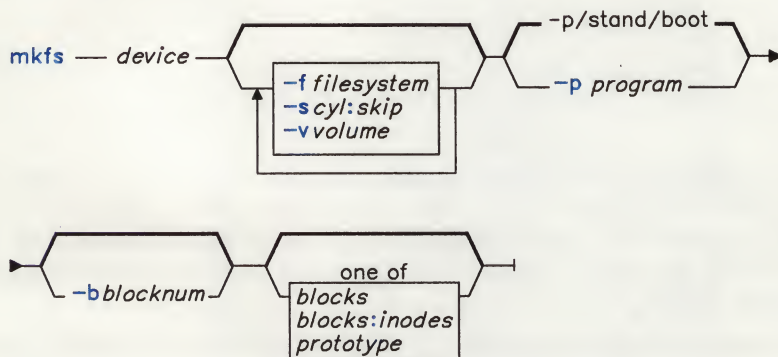
# mkfs

# mkfs

## Purpose

Makes a file system.

## Syntax



OL805364

## Description

The **mkfs** command makes new file systems. **mkfs** initializes the volume label and file system label, start-up block, bad-block list, and interleaves the free list in accordance with the flags or with defaults found in the `/etc/filesystems` file.

The **mkfs** command creates the new file system on the *device* specified on the command line. *device* can be a block device name, raw device name, or file system name. If it is a file system name, **mkfs** uses this name as the *file system* and uses the following parameters from the applicable stanza in `/etc/filesystems`:

<b>dev</b>	Device name.
<b>cyl</b>	See the following <b>-s</b> flag.
<b>skip</b>	See the following <b>-s</b> flag.
<b>vol</b>	Volume ID.
<b>bad</b>	List of bad blocks separated by commas.
<b>size</b>	File system size.
<b>boot</b>	Program to be installed in startup block.

## File System Size

You can specify the size of a new file system in the following way:

- On the command line
- In the *prototype* file
- In the */etc/filesystems* entry for the given file system.

If the size is not specified in any of these places, **mkfs** takes it from the **devinfo** structure for the block device associated with the file system being generated. (See the **ioctl** system call and the **devinfo** file in *AIX Operating System Technical Reference*.) The size provided in the **devinfo** structure is the maximum size of the file system in any case. A size specification on the command line overrides any defaults found in the **devinfo** structure or in */etc/filesystems*.

## Prototype Files

To initialize the contents of a new file system in accordance with a prototype, specify the name of a *prototype* file on the command line. The **proto** command can be used to construct prototype files from existing file systems.

The *prototype* file contains tokens separated by spaces or new-line characters. The first token is the name of a file to be copied onto block 0 as the bootstrap program. The second token is a number specifying the size of the created file system. Typically it is the number of blocks on the device, perhaps diminished by space for paging. The next token is the number of i-nodes in the i-list. (**mkfs** rounds this to fill out the appropriate number of blocks.) The next set of tokens contains the specifications for the root file. File specifications consist of tokens giving the mode, the user name, the group name, and the initial contents of the file. The syntax of the contents field depends on the mode.

The mode token for a file is a six-character string. The first character specifies the type of the file. (The characters **-**, **b**, **c**, and **d** specify regular, block special, character special, and directory files, respectively.) The second character must be either **u** or **-**. If **u** is used, the set-user-ID mode is specified; if **-** is used, the set-user-ID mode is not specified. The third character must be either **g** or **-** for specifying the set-group-ID mode. The rest of the mode is a three-digit octal number giving the owner, group, and other read, write, execute permissions (see "**chmod**" on page 160).

Two decimal number tokens come after the mode. They specify the user and group names of the owner of the file.

If the file is a regular file, the next token is a path name from which the contents and size are copied.

If the file is a block or character special file, two decimal number tokens follow, which give the major and minor device numbers.



## mkfs

---

If the file is a directory, **mkfs** makes the entries **.** (dot) and **..** (dot dot) and then recursively reads a list of names and file specifications for the entries in the directory. The scan is ended with the token **\$** (dollar sign).

### Flags

<b>-bblocknum</b>	When present, specifies the number of blocks allocated to file <b>i-node1</b> which is automatically created.
<b>-ffilesystem</b>	Specifies the file system label for the new file system. This can be up to 6 bytes.
<b>-pprogram</b>	Specifies the name of a program to be installed in block 0 of the new file system. The default bootstrap program is <b>/stand/boot</b> .
<b>-scyl:skip</b>	Specifies an interleaving of the free list. (Interleaving the free list can improve the speed of disk I/O.) <i>cyl</i> is the number of blocks per cylinder, and <i>skip</i> is the number of blocks to skip.
<b>-vvolume</b>	Specifies the volume label for the new file system. This can be up to 6 bytes.
<b>blocks[:inodes]</b>	A size specification where <i>blocks</i> is the number of 512-byte blocks in the file system. When <i>inodes</i> is specified, it determines the number of i-nodes on the system. If <i>inodes</i> is not specified, a number suitable for the size of the file system is used. The number of i-nodes is rounded up so that the i-node area occupies an integral number of blocks.

### Examples

1. To create an empty file system on a diskette:

```
mkfs /dev/fd0
```

2. To specify volume and file system names for a new file system:

```
mkfs /dev/fd0 -fWORKFS -vVOL001
```

This creates an empty file system on the diskette, giving it the volume serial VOL001 and file system name WORKFS.

### Related Information

The following command: **"fsck, dfsc"** on page 445.

The **ioctl** system call and the **devinfo**, **dir**, **filesystems**, and **fs** files in *AIX Operating System Technical Reference*.

---

# mknod

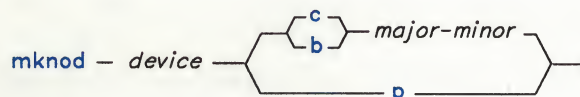
---

## Purpose

Creates a special file.

## Syntax

`mknod - device { c b } major-minor p`



OL805146

## Description

The **mknod** command makes a directory entry and corresponding i-node for a special file. The first parameter is the name of the entry *device*. Select a name that is descriptive of the device.

The **mknod** command has two forms. In the first case, the second argument is **b** or **c**. The **b** argument indicates that the special file is a block-oriented device (disk, diskette, tape). The **c** argument indicates that it is a character-oriented device (other devices). The last two parameters are numbers specifying the *major* device, which helps the operating system find the device driver code, and the *minor* device, that is, the unit drive, or line number, which may be either decimal or octal.

The assignment of major device numbers is specific to each system. Device numbers are determined by examining the system source file **conf.c**.

**Note:** If you change the contents of **conf.c** to add a device driver, you must rebuild the operating system. See the discussion of device drivers in *AIX Operating System Programming Tools and Interfaces* and in *AIX Operating System Technical Reference*.

The second form of **mknod** is used to create FIFOs (named pipes). The **p** flag after *device* indicates that you are creating a named pipe. See the *AIX Operating System Technical Reference* for an explanation of FIFOs and named pipes.

## mknod

---

### Example

To create the special file for a new diskette drive:

```
mknod /dev/fd2 b 1 2
```

This creates the special file **/dev/fd2**, which is a block special file with major device number 1 and minor device number 2.

### Related Information

The **mknod** file and device driver description in *AIX Operating System Technical Reference*.

The discussion of device drivers in *AIX Operating System Programming Tools and Interfaces*.

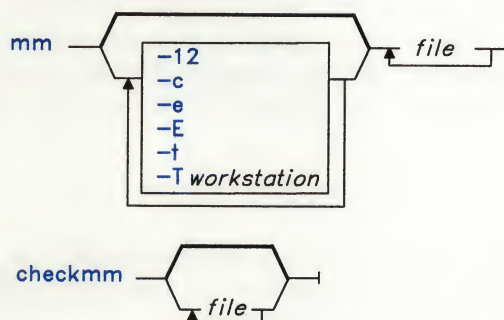


## mm, checkmm

### Purpose

Displays or checks documents formatted with Memorandum Macros.

### Syntax



OL805039

### Description

Using the **nroff** command and the Memorandum Macro text-formatting package (MM), the **mm** command writes *files* to standard output. If you specify a - (minus) instead of any *files*, **mm** reads standard input. Do not specify both file names and standard input on the command line.

The **mm** command has flags to specify preprocessing by the **tbl** and/or **eqn** commands and postprocessing by various work station oriented-output filters. It generates the proper pipelines, and the required arguments for **nroff** and MM, depending on the flags selected, creates the required pipelines.

The **checkmm** command is a program for checking the contents of the named *files* for errors in the use of MM and some **eqn** and **neqn** constructions. The program skips all directories, and if you do not specify a file, **checkmm** reads standard input.

### Notes:

1. Use the **-olist** argument of **nroff** to specify ranges of pages to be output. Note, however, that invoking **mm** with one or more of the **-e**, **-t**, and **-** minus arguments together with **nroff -olist** may cause a harmless broken pipe diagnostic if the last page of the document is not specified in *list*.
2. The **mm** command calls **nroff** with the **-h** flag. With this flag, **nroff** assumes that the work station has tabs set every eight character positions.
3. If you use the **-s** flag of **nroff** (to stop between pages of output), use a line feed (rather than **Enter** or a new-line character) to restart the output. The **-s** flag of **nroff** does not work with the **-c** flag of **mm** or if **mm** automatically calls the **col** command.
4. If you provide inaccurate information to **mm** about the kind of work station its output is to be printed on, you will get unsatisfactory results; however, if you are redirecting output to a file, use the **-T37** flag and then use the appropriate work station filter when you actually print the file.

## Flags

Any flags on the command line not listed below are passed to **nroff** or to **MM**, as appropriate. The flags can occur in any order, but they must come before *file*. To obtain a list of **mm** flags, enter the command name with no arguments.

- c** Invokes the **col** command. Note that **col** is invoked automatically by **mm** unless *workstation* (the **-T** flag parameter) is one of the following:
- 300
  - 300s
  - 450
  - 37
  - 4000a
  - 382
  - 4014
  - tek
  - 1620
  - X
- e** Invokes the **neqn** command.
- E** Invokes the **-e** flag of **nroff**.
- t** Invokes the **tbl** command.
- Tworkstation** Uses work station specification *workstation*. For a list of recognized values for *workstation*, enter:
- help term1

By default, **mm** uses the value of the shell variable **\$TERM** from the environment as the value of *workstation*. If **\$TERM** is not set **mm** uses **lp**. If several work station types are specified, the last one listed takes effect.

-12

Uses 12-pitch font. This may be used when **\$TERM** is set to one of **300**, **300s**, **450**, or **1620**. (The pitch switch on the DASI 300 and 300s work stations must be manually set to 12 if this flag is used.)

## Related Information

The following commands: “**col**” on page 179, “**env**” on page 393, “**eqn**, **neqn**, **checkeq**” on page 395, “**greek**” on page 499, “**mmt**, **checkmm**” on page 666, “**nroff**, **troff**” on page 709, and “**tbl**” on page 1053.

The **profile** file and the **eqnchar**, **mm**, and **term** miscellaneous facilities in *AIX Operating System Technical Reference*.

The discussion of **mm** in *Text Formatting Guide*.



## mmt

---

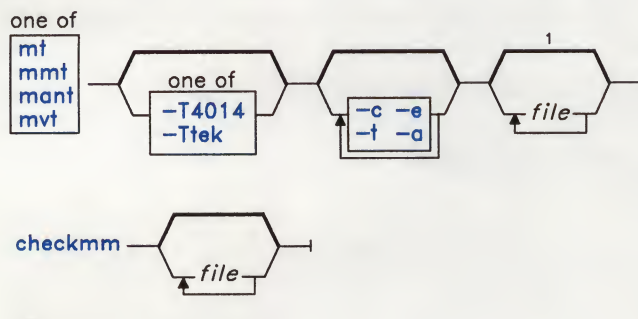
## mmt, checkmm

---

### Purpose

Typesets documents, manual pages, view graphs, and slides.

### Syntax



<sup>1</sup> If no files are given, these commands will display their flags.

OL805092

### Description

These commands are similar to the **mm** command, except they typeset their input via **troff** as opposed to formatting it via **nroff**. The **mvt**, **mt**, and **mant** commands are links to **mmt**. **mmt** uses the MM Macro Package (see **mm** in *AIX Operating System Technical Reference*), **mvt** uses the macro package for view graphs and slides (see **mv** in *AIX Operating System Technical Reference*), **mant** uses the manual page macros, and **mt** does not use a macro package.

These commands have flags to specify preprocessing by **tbl**, **cw**, or **eqn**. **mmt** generates the proper pipelines and the required arguments for **troff** and for the macro package used, depending on the flags selected. These commands read standard input if you specify a - (minus) instead of any file names.

The **checkmm** command can be used to check the input to **mmt**.

If the input contains a **troff** comment line consisting solely of the string `'\" x` (single quotation mark, backslash, double quotation mark *x*), where *x* is any combination of the three letters **c**, **e**, and **t** and where there is exactly one blank between the double quotation mark and *x*, then the input will be processed through the appropriate combination of **cw**, **eqn**, and **tbl**, respectively, regardless of the command-line arguments.

**Note:** Use the **-olist** argument of **troff** to specify ranges of pages to be output. Note, however, that calling these commands with one or more of the **-c**, **-e**, **-t**, and **-** arguments together with **troff -olist** may cause a harmless **broken pipe** diagnostic if the last page of the document is not specified in *list*.

## Flags

Flags other than the ones listed below are passed to **troff** or to the macro package, as appropriate. All flags must appear before the *file* names. If you do not provide any arguments, these commands print a list of their flags.

- a**        Invokes the **-a** flag of **troff**.
- c**        Preprocesses the input files with **cw**.
- e**        Preprocesses the input files with **eqn**.
- t**        Preprocesses the input files with **tbl**.
- T4014**
- Ttek**    Directs the output to a Tektronix 4014 work station via the **tc** command.

## Related Information

The following commands: “**env**” on page 393, “**eqn**, **neqn**, **checkeq**” on page 395, “**mm**, **checkmm**” on page 663, “**tbl**” on page 1053, “**tc**” on page 1056, and “**troff**” on page 710.

The **profile** file and the **environ**, **mm**, and **mv** miscellaneous facilities in *AIX Operating System Technical Reference*.

**moo**

---

**moo**

---

## Purpose

Plays a number-guessing game.

## Syntax

`/usr/games/moo` —

OL805231

## Description

The **moo** command picks a random four-digit decimal number with nonrepeating digits. You guess four digits and score a “cow” with a correct digit in an incorrect position and a “bull” with a correct digit in a correct position. The game continues until you guess the number.

To quit the game, press INTERRUPT (**Alt-Pause**) or END OF FILE (**Ctrl-D**).

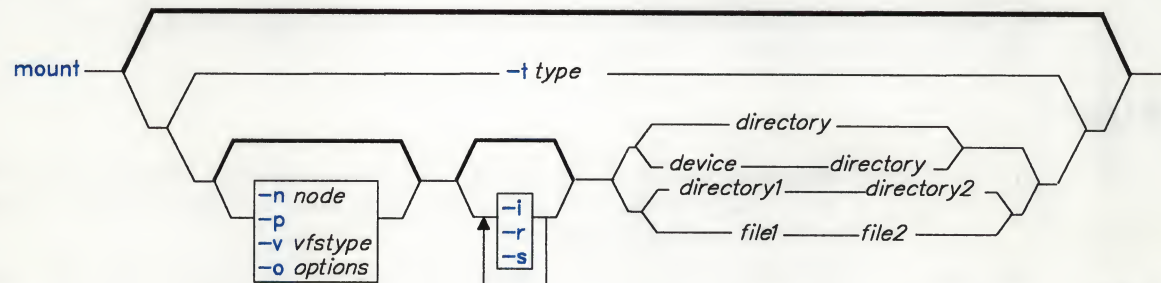


# mount

## Purpose

Makes a file system available for use.

## Syntax



OL805467

## Description

The **mount** command instructs the operating system to make a file system available for use. In addition, you can use **mount** to build other file trees made up of directory and file mounts. In the case of file system mounts, the **mount** command mounts the specified *device* on the specified *directory*. After **mount** has finished, *directory* becomes the root of the newly mounted file system.

Any user can issue a **mount** *directory1 directory2* or **mount** *file1 file2* command if that user has search or write permission to the directory or file to be mounted over (*directory2* or *file2*). Members of the system group can also do any mount described in the `/etc/filesystems` file (**mount** *directory*). Users operating as superusers can issue any **mount** command.

If you enter the **mount** command without arguments, it writes to standard output the mounted file systems, their locations, and their mount options.

If you specify only a *directory* name, **mount** takes it to be the name of the directory or file on which a file system, directory, or file is usually mounted (as defined in the `/etc/filesystems` file). **mount** looks up the associated device, directory, or file and mounts it. This is the most convenient way of using the **mount** command, as it does not require you to remember what is normally mounted on a directory or file.

## mount

---

The `/etc/filesystems` file should include a *stanza* for each mountable file system, directory, or file. This stanza should specify at least the name of the file system and either the device on which it resides or the directory name. If the stanza includes a *mount attribute*, the `mount` command uses the associated values. It recognizes five values in the mount attribute: **true**, **false**, **removable**, **inherit** and **read-only** (see the `filesystems` file in *AIX Operating System Technical Reference* for a description of these mount attributes.) The command `mount all` causes all file systems with the attribute `mount=true` to be mounted in their normal places. This command is typically used during system initialization.

If you are operating with superuser authority, you can mount a file system arbitrarily by naming both a *device* and a *directory* on the command line. `mount` takes *device* to be the name of the block device special file and *directory* to be the directory on which it should mount the file system.

## Flags

- i** Requests an inherited mount. This option is only valid for Distributed Services mounts. (For information on inherited mounts, see *Managing the AIX Operating System*.)
- n node** Specifies the node that holds the directory to be mounted. If you specify this option without specifying the directory to be mounted, `mount` displays a list of all mounts issued at *node*, if the nodes are connected with Distributed Services.
- o** Sets options for a hard or soft mount in the Network File System environment. The following options can be used:
  - bg** Mount is attempted in background if first attempt fails.
  - fg** Mount is attempted in foreground if first attempt fails.
  - retry = n** Mount is attempted *n* times.
  - rsize = n** Sets read buffer size to *n* bytes.
  - wsiz e = n** Sets write buffer size to *n* bytes.
  - timeo = n** Sets NFS timeout period to *n* tenths of a second.
  - retrans = n** Sets the number of NFS transmissions to *n*.
  - port = n** Sets server IP port number to *n*.
  - soft** Error is returned if server does not respond.
  - hard** Request is retried until server responds.
  - intr** Allows keyboard interrupts on hard mounts.
  - rw** Mounted file is read/write accessible.



**ro** Mounted file is read-only.

The default options and their values (if any) are as follows: **fg**, **retry** = 10000, **rsize** = 8192, **wsizes** = 8192, **timeo** = 7, **retrans** = 3, **port** = NFS\_PORT (a system-specified constant), **hard**, and **rw**.

**Note:** Options you enter on the command line should be separated only by a comma, not a comma and a space.

- p** Mounts a file system as a removable file system. While there are open files on it, a removably mounted file system behaves the same as a normally mounted file system. However, when there are no open files (and no process has a current directory on the file system), all of the file system's disk buffers are written to the medium, and the operating system "forgets" the structure of the file system. This allows you to remove and reinsert media such as diskettes without issuing a **mount** or **umount** command each time. Use this flag only for diskette mounts.
- r** Mounts a file system as a read-only file system, regardless of the specification in **/etc/filesystems**.
- s** This flag is for backwards compatibility only.
- t type** Mounts all stanzas in **/etc/filesystems** that contain **type** = *type* and are not mounted. (*type* is a string value, such as remote.)
- v vfstype** Mounts the file system using the specified file system type, such as **aix**, **ds**, or **nfs**. If no **vfstype** is specified, the defaults set in the **/etc/vfs** file are used.

## Examples

1. To list the file systems that are mounted:

**mount**

nodename	mounted	mounted over	vfs	date	options
-----	-----	-----	---	-----	-----
-	/dev/hd0	/	aix	Dec 17 08:04	rw
-	/dev/hd6	/vrm	aix	Dec 17 08:05	ro
-	/dev/hd1	/u		Dec 17 08:06	rw
-	/dev/hd2	/usr		Dec 17 08:06	rw
-	/dev/hd3	/tmp		Dec 17 08:06	rw
darlene	/usr	/usr	ds	Dec 17 10:44	rw

For each file system, **mount** lists the node name, the device name, the name under which it is mounted, the access permitted (read only or read/write), and the date and time it was mounted.



## mount

---

2. To mount a diskette:

```
mount /dev/fd0 /diskette0
```

This mounts a diskette (/dev/fd0) onto the directory /diskette0. A file system must already exist on the diskette, and the directory /diskette0 must already exist. To access a file on the diskette, use a path name that begins with /diskette0. For example, to access prog.c use /diskette0/prog.c.

**Warning:** Be sure that the current directory is not still on the diskette when you remove it from the drive, or you may lose some of your data.

3. To mount a write-protected diskette:

```
mount -r /dev/fd0 /diskette0
```

This mounts the diskette on /diskette0 as a read-only file system. This tells the operating system not to update file access times, which would cause errors with a write-protected diskette.

4. To mount a default file system:

```
mount /diskette0
```

This mounts the device that is usually mounted on /diskette0, which is determined by information in the file /etc/filesystems.

5. To mount all default file systems:

```
mount all
```

This mounts all standard file systems in /etc/filesystems marked mount = true.

6. To mount a remote directory:

```
mount -n nodeA /u/tom /u/tom
```

This mounts the remote **nodeA** directory /u/tom onto the local node directory /u/tom. It assumes the default remote **vfs\_type** (which must be defined in /etc/vfs).

7. To mount a file or directory from the /etc/filesystem file with a specific type:

```
mount -t remote
```

This mounts all files or directories in the /etc/filesystems file that have a stanza that contains the attribute type = remote.

## Files

/etc/filesystems	Descriptions of mountable file systems.
/etc/vfs	Descriptions of file system types.

## Related Information

The following command: “**umount, unmount**” on page 1112.

The **mount**, **mntctl**, **umount**, and **vmount** system calls and the **vfs** and **filesystems** files in *AIX Operating System Technical Reference*.

The discussion of distributed services, code service, NFS, and the overview of international character support in *Managing the AIX Operating System*.

## mountd

---

## mountd

---

### Purpose

Answers NFS mount requests.

### Syntax

`/usr/etc/rpc.mountd` —|

OL805506

### Description

The **mountd** daemon answers file system mount requests. It reads the file `/etc/exports` to determine which file systems are available to which machines and users in the network.

This daemon also identifies clients that have file systems mounted. Use the **showmount** command to display this information.

The **inetd** daemon invokes the **mountd** daemon.

---

#### Japanese Language Support Information

If Japanese Language Support is installed on your system, this command is not available.

---

### Files

`/etc/exports`  
`/etc/inetd.conf`

### Related Information

The following command: “**showmount**” on page 945.

The **inetd** daemon in *IBM RT Interface Program for use with TCP/IP*.

The NFS chapter in *Managing the AIX Operating System*.



---

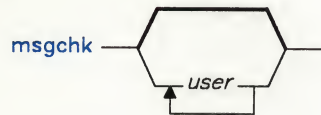
# msgchk

---

## Purpose

Checks for messages.

## Syntax



`msgchk -help`

AJ2FL231

## Description

The **msgchk** command is used to check mail drops for messages waiting to be received. **msgchk** is part of the MH (Message Handling) package and can be used with other MH and AIX commands.

The **msgchk** command checks all mail drops belonging to the specified user IDs and reports which mail drops contain messages that have not been received. **msgchk** also indicates whether it appears you have already seen these messages. If you do not specify a *user* argument, **msgchk** checks the current user's mail drops.

## Flag

**-help** Displays help information for the command.

## Files

<code>\$HOME/.mh-profile</code>	The MH user profile.
<code>/usr/lib/mh/mtstailor</code>	The MH tailor file.
<code>/usr/mail/\$USER</code>	The location of the mail drop.

### Related Information

The MH command “**inc**” on page 518.

The **mh-mail** and **mh-profile** files in *AIX Operating System Technical Reference*.

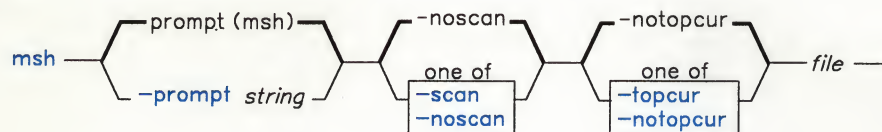
“Overview of the Message Handling Package” in *Managing the AIX Operating System*.

# msh

## Purpose

Creates an MH shell.

## Syntax



msh — -help —

AJ2FL232

## Description

The **msh** command is used to work with messages stored in a packed format. **msh** is part of the MH (Message Handling) package and can be used with other MH and AIX commands.

The **msh** command performs a modified subset of MH commands on messages stored in packed format. **msh** prompts you to enter one of the following MH commands, and continues to prompt you for commands until you press **END OF FILE** or enter **quit**:

<b>ali</b>	<b>burst</b>	<b>comp</b>	<b>dist</b>
<b>folder</b>	<b>forw</b>	<b>inc</b>	<b>mark</b>
<b>mhmail</b>	<b>msgchk</b>	<b>next</b>	<b>packf</b>
<b>pick</b>	<b>prev</b>	<b>refile</b>	<b>repl</b>
<b>rmm</b>	<b>scan</b>	<b>send</b>	<b>show</b>
<b>sortm</b>	<b>whatnow</b>	<b>whom</b>	

You can also enter **help** to display a brief overview.



## msh

---

### Flags

<b>-help</b>	Displays help information for the command.
<b>-noscan</b>	Does not scan unseen items.
<b>-notopcur</b>	Makes the current message track the center line of the <b>vmh</b> scan window when <b>msh</b> is invoked from <b>vmh</b> . This flag is the default.
<b>-prompt <i>string</i></b>	Prompts for <b>msh</b> commands with the specified string.
<b>-scan</b>	Scans unseen items.
<b>-topcur</b>	Makes the current message track the top line of the <b>vmh</b> scan window when <b>msh</b> is invoked from <b>vmh</b> .

### Profile Entries

<b>fileproc:</b>	Specifies the program used to refile messages.
<b>Msg-Protect:</b>	Sets the protection level for your new message files.
<b>Path:</b>	Specifies your <i>user-mh-directory</i> .
<b>showproc:</b>	Specifies the program used to show messages.

### Files

<b>\$HOME/.mh-profile</b>	The MH user profile.
<b>/usr/lib/mh/mtstailor</b>	The MH tailor file.

### Related Information

The following MH commands: “**ali**” on page 48, “**burst**” on page 129, “**comp**” on page 185, “**dist**” on page 336, “**folder**” on page 429, “**forw**” on page 438, “**inc**” on page 518, “**mark**” on page 637, “**mhmail**” on page 646, “**msgchk**” on page 675, “**next**” on page 694, “**packf**” on page 733, “**pick**” on page 748, “**prev**” on page 765, “**refile**” on page 817, “**repl**” on page 821, “**rmm**” on page 841, “**scan**” on page 871, “**send**” on page 893, “**show**” on page 942, “**sortm**” on page 965, “**whatnow**” on page 1215, “**whom**” on page 1222.

The **mh-alias**, **mh-format**, **mh-mail**, and **mh-profile** files in *AIX Operating System Technical Reference*.

“Overview of the Message Handling Package” in *Managing the AIX Operating System*.

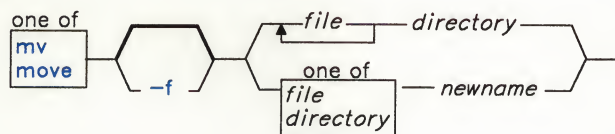
---

**mv**


---

**Purpose**

Moves files.

**Syntax**

OL805010

**Description**

**Warning:** The **mv** command may overwrite many files if you do not ensure that the file path names you are specifying do not already exist.

The **mv** (**move**) command moves files from one directory to another, or it renames a file or directory. If you move a *file* to a new *directory*, it retains the base file name. When you move a file, all links to other files remain intact, except when you move it to a different file system.

You can only rename a *directory* with **mv**; you cannot move it. Both *directory* and *newname* must have the same parent. All files in *directory* are moved to a newly-created directory *newname* under the same file names.

When you use **mv** to rename a *file*, then *newname* can specify either a new file name or a new directory path name. If moving the file would overwrite an existing write-protected file and if standard input is a work station, **mv** displays the permission code of the file to be overwritten and reads one line from standard input. If the line begins with y, the move takes place and the file is overwritten. If not, **mv** does nothing with the *file*.

---

**Japanese Language Support Information**

An affirmative response in Japanese Language Support matches one of the elements in the environment variable **YESSTR**.

---

**Note:** If the *file* is on different file system than *directory*, **mv** must copy the *file* to the new file system and delete the original. In this case, the owner name becomes that of the user, and all links to other files are lost.

## Flags

- f** Does not prompt before removing a write-protected file.

## Examples

1. To rename a file:

```
mv appendix apndx.a
```

This renames `appendix` to `apndx.a`. If a file named `apndx.a` already exists, its old contents are replaced with those of `appendix`.

2. To rename a directory:

```
mv book manual
```

This renames `book` to `manual`. If a directory named `manual` already exists, then an error message is displayed.

3. To move a file to another directory and give it a new name:

```
mv intro manual/chap1
```

This moves `intro` to `manual/chap1`. The name `intro` is removed from the current directory, and the same file appears as `chap1` in the directory `manual`.

4. To move a file to another directory, keeping the same name:

```
mv chap3 manual
```

This moves `chap3` to `manual/chap3`.

**Note the difference:** Examples 1 and 3 name two files, Example 2 names two existing directories, and Example 4 names a file and a directory.

5. To move several files into another directory:

```
mv chap4 jim/chap5 /u/manual
```

This moves `chap4` to `/u/manual/chap4` and `jim/chap5` to `/u/manual/chap5`.

6. To use **mv** with pattern-matching characters:

```
mv manual/* .
```

This moves all files in the directory `manual` into the current directory (`.`), giving them the same names they had in `manual`. This also empties `manual`. Note that you must type a space between the star and the period.



## Related Information

The following commands: “**chmod**” on page 160, “**ln**” on page 581, and “**rm**” on page 833.

The **rename** system call in *AIX Operating System Technical Reference*.

The discussion of Japanese Language Support in *Japanese Language Support User's Guide*.

## **mvdir**

---

## **mvdir**

---

### **Purpose**

Moves (renames) a directory.

### **Syntax**

**mvdir** — *directory1* — *directory2* —

OL805137

### **Description**

The **mvdir** command renames directories within a file system. To use **mvdir**, you must have write permission to *directory1* and *directory2* and to the parent directories of *directory1* and *directory2*. The *directory1* parameter must name an existing directory. If *directory2* does not exist, *directory1* is moved to *directory2*. If *directory2* exists, *directory1* becomes a subdirectory of *directory2*. Neither directory can be a subset of the other.

**Note:** *directory1* and *directory2* may be the names of files. If *directory2* is a file name, it is replaced with *directory1*.

### **Example**

To rename or move a directory to another location:

```
mvdir  appendixes  manual
```

If *manual* does not exist, then this renames the directory *appendixes* to *manual*. You can also rename a directory with the **mv** command.

If a directory named *manual* already exists, this moves *appendixes* and its contents to *manual/appendixes*. In other words, *appendixes* becomes a subdirectory of *manual*.

### **Related Information**

The following commands: “**mkdir**” on page 657 and “**mv**” on page 679.

The **rename** system call in *AIX Operating System Technical Reference*.

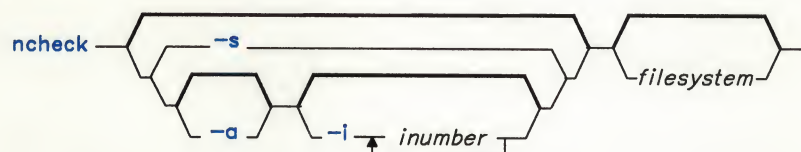
# ncheck

---

## Purpose

Generates path names from i-numbers.

## Syntax



OL805196

## Description

The **ncheck** command without any flags writes to standard output the path name and i-number list for all files in *filesystem*.

If you specify an invalid file system, the ?? in the name stands for the parent of a file system that does not have a parent. Path names beginning with ... (dot dot dot) indicate a loop.

---

### Japanese Language Support Information

This command has not been modified to support Japanese characters.

---

## Flags

- |                         |   |
|-------------------------|---|
| <b>-a</b>               | Lists includes the file names . (dot) and .. (dot dot).   |
| <b>-i inumber . . .</b> | Lists only the file specified by <i>inumber</i> .         |
| <b>-s</b>               | Lists only special files and files with set-user-ID mode. |

## Examples

- To list the i-number and path name of each file in the default file systems:  
ncheck



## ncheck

---

2. To list all the files in a specified file system:

```
ncheck -a /
```

This lists the i-number and path name of each file in the root file system (/), including the . (dot) and .. (dot-dot) entries in each directory (-a).

3. To list the name of a file when you know its i-number:

```
ncheck -i 690 357 280 /diskette0
```

This lists the i-number and path name for every file in the file system **/diskette0** with i-numbers of 690, 357, or 280. If a file has more than one link, all of its path names are listed.

4. To list special and set-user-ID files:

```
ncheck -s /
```

This lists the i-number and path name for every file in the root file system that is a special file (also called a **device file**) or that has set-user-ID mode enabled.

## Related Information

The following commands: “**fsck**, **dfsck**” on page 445 and “**sort**” on page 958.

## ndtable

---

### Purpose

Accesses the Distributed Services Node Table.

### Syntax

`ndtable` —

OL805470

### Description

The **ndtable** command lets you build, examine, or change the Distributed Services Network Node Table. Only members of the system group or users operating with superuser authority can use **ndtable** to change the state of the table (see “su” on page 1026). Other users can use **ndtable** to browse through the table.

**Note:** If your system crashes when a user is running **ndtable** to view or update a remote machine’s table, you will get system error message 000-002 the next time you try to access the table. If this happens, remove the contents of the **/etc/profsvcs/NID** directory (where *NID* is the numeric node ID of the remote machine), and retry the **ndtable** command.

### Related Information

“Getting Started With Distributed Services Configuration Menus” in *Managing the AIX Operating System*.

## newform

---

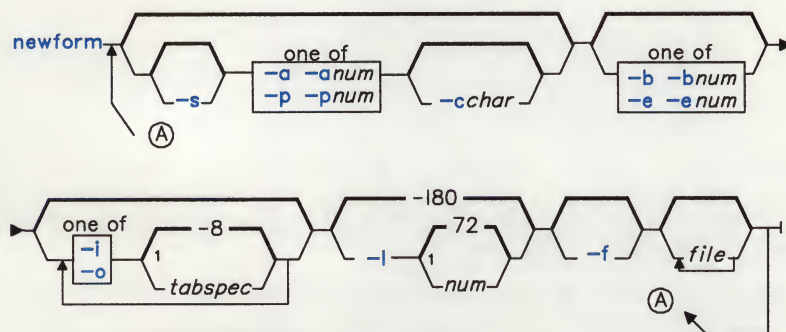
## newform

---

### Purpose

Changes the format of a text file.

### Syntax



<sup>1</sup> Do not put a blank between these items.

OL805197

### Description

The **newform** command takes lines from *file* (standard input by default), and writes the formatted lines to standard output. Lines are reformatted in accordance with the command line flags in effect.

Except for **-s**, command line flags can appear in any order, can be repeated, and can be mixed with the *file* parameter. Command line flags are processed in the order specified. In other words, flag sequences like **-e15 -l60** yield results different from **-l60 -e15**. Flags are applied to all *files* on the command line.

An exit value of 0 indicates normal execution, a 1 indicates an error.



**Notes:**

1. The **newform** command normally only keeps track of physical characters; however, for the **-i** and **-o** flags, **newform** keeps track of backspaces in order to line up tabs in the appropriate logical columns.
2. The **newform** command does not prompt you if a *tabspec* is to be read from the standard input (by use of **-i--** or **-o --**).
3. If the **-f** flag is used and the last **-o** flag specified was **-o--** and was preceded by either a **-o--** or a **-i--**, the tab specification format line will be incorrect.

---

**Japanese Language Support Information**

This command has not been modified to support Japanese characters.

---

**Flags**

- a[num]** Adds *num* characters to the end of the line when the line length is less than the effective line length (see the **-c** and **-p** flags in this section).
- b[num]** Truncates *num* characters from the beginning of the line when the line length is greater than the effective line length (see **-lnum**). The default action truncates the number of characters necessary to obtain the effective line length. If you specify **-b** with no *num*, the default takes effect. This flag can be used to delete the sequence numbers from a COBOL program as follows:
- ```
newform -ll-b7 file-name
```
- The **-ll** must be used to set the effective line length shorter than any existing line in the file so that the **-b** flag is activated.
- c[char]** Changes the prefix/add character to *char*. Default character for *char* is a space.
- e[num]** Same as **-bnum** except that characters are truncated from the end of the line.
- f** Writes the tab specification format line to standard output before any other lines are written. The tab specification format line displayed corresponds to the format specified in the *last* **-o** flag. If no **-o** flag is specified, the line displayed contains the default specification of **-8**.
- i[tabspec]** Replaces all tabs in the input with the number of spaces specified by *tabspec*. *tabspec* recognizes all tab specification forms described in "tabs" on page 1041. If you specify a -- (minus minus) for the value of *tabspec*, **newform** assumes that the tab specification can be found in the first line read from standard input (see **fspec** in *AIX Operating System Technical Reference*). The default *tabspec* is **-8**. A *tabspec* of **-0** expects no tabs; if any are found, they are treated as **-1**.

## newform

---

- l[*num*]** Sets the effective line length to *num* characters. If *num* is not entered, **-l** defaults to 72. The default line length without the **-l** flag is 80 characters. Note that tabs and backspaces are considered to be one character (use **-i** to expand tabs to spaces).
- o[*tabspec*]** Replaces spaces in the input with a tab in the output, according to the tab specifications given. The default *tabspec* is -8. A *tabspec* of -0 means that no spaces are converted to tabs on output.
- p[*num*]** Prefixes *num* characters (see **-cchar**) to the beginning of a line when the line length is less than the effective line length. The default action is to prefix the number of characters that are necessary to obtain the effective line length.
- s** Removes leading characters on each line up to the first tab and places up to eight of the removed characters at the end of the line. If more than eight characters (not counting the first tab) are removed, the eighth character is replaced by an \* (asterisk) and any characters to the right of it are discarded. The first tab is always discarded.

The removed characters are saved internally until all other flags specified are applied to that line. The characters are then added at the end of the processed line.

For example, to convert a file with leading digits, one or more tabs, and text on each line, to a file beginning with the text, all tabs after the first expanded to spaces, padded with spaces out to column 72 (or truncated to column 72), and the leading digits placed starting at column 73, the command would be as follows:

```
newform -s -i -l -a -e file-name
```

The **newform** command displays an error message and stops if this flag is used on a file without a tab on each line.

## Related Information

The following commands: “**tabs**” on page 1041 and “**csplit**” on page 252.

The **fspec** file in *AIX Operating System Technical Reference*.



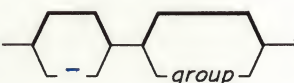
## newgrp

---

### Purpose

Changes your primary group identification.

### Syntax

`newgrp` 

OL805198

### Description

The **newgrp** command changes your **primary group** identification to *group*. **newgrp** recognizes only group names, not group ID numbers. Without an argument, it changes your primary group to the one specified in the `/etc/passwd` file.

If the group has a password and you do not or if the group has a password and you are not listed in the `/etc/group` file as a member, then **newgrp** asks you for the group password. (The use of group passwords is not encouraged because, by their very nature, they encourage poor security practices.)

**Note:** Any active user-generated shell will be terminated when **newgrp** is used.

---

#### Japanese Language Support Information

This command has not been modified to support Japanese characters.

---

### Flag

- Changes the environment to the login environment of the new group.

### Examples

1. To change the primary group ID of the current shell session to **admin**:  
`newgrp admin`
2. To change the primary group ID back to your original login group:  
`newgrp`



## **newgrp**

---

### **Files**

|             |                                   |
|-------------|-----------------------------------|
| /etc/group  | Group file; contains group IDs.   |
| /etc/passwd | Password file; contains user IDs. |

### **Related Information**

The following commands: “**login**” on page 584 and “**users, adduser**” on page 1129.

The **group** and **passwd** files in *Installing and Customizing the AIX Operating System*.

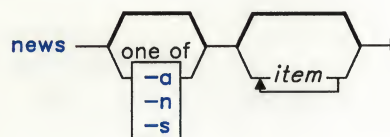
## news

---

### Purpose

Writes system news items to standard output.

### Syntax



OL805199

### Description

The **news** command keeps you informed of news concerning the system. Each news *item* is contained in a separate file in directory **/usr/news**. Anyone having read/write permission to this directory can create a news file.

If you run the **news** command without any flags, it displays every current file in **/usr/news**, showing the most recent first. Or you can specify the *items* you want displayed.

Each file is preceded by an appropriate header. To avoid reporting old news, **news** stores a currency time. **news** considers your currency time to be the modification time of the file named **\$HOME/.news\_time**. Each time you read the news, the modification time of this file changes to that of the reading. Only news item files posted after this time are considered current.

Pressing INTERRUPT (**Alt-Pause**) during the display of a news item stops the display of that item and starts the next. Pressing INTERRUPT (**Alt-Pause**) again ends **news**.

Most users run **news** each time they log in by including the line:

```
news -n
```

in their **\$HOME/.profile** file or in the system's **/etc/profile**.

### Flags

- a Displays all news items, regardless of the currency time. The currency time does not change.

## news

---

- n Reports the names of current news items without displaying their contents. The currency time does not change.
- s Reports the number of current news items without displaying their names or contents. The currency time does not change.

## Examples

1. To display the items that have been posted since you last read the news:

```
news
```

2. To display all the news items:

```
news -a | pg
```

This displays all the news items a page at a time (l pg) whether or not you have read them yet.

3. To list the names of the news items that you have not read yet:

```
news -n
```

Each name is a file in the directory **/usr/news**.

4. To display specific news items:

```
news newusers services
```

This displays news about newusers and services, which are names listed by **news -n**.

5. To display the number of news items that you have not read yet:

```
news -s
```

6. To post news for everyone to read:

```
cp schedule /usr/news
```

This copies the file schedule into the system news directory, **/usr/news**, to create the file **/usr/news/schedule**. To do this you must have write permission for **/usr/news**.

## Files

```
/etc/profile  
/usr/news/*  
$HOME/.news_time
```



## Related Information

The following command: **pg** on page 744.

The **profile** file and **environ** special facility in *AIX Operating System Technical Reference*.

**next**

---

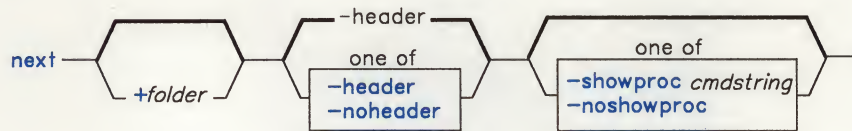
**next**

---

## Purpose

Shows the next message.

## Syntax



**next** — **-help** —

AJ2FL159

## Description

The **next** command is used to display the next message in a folder. **next** is equivalent to the **show** command with **next** specified as the message. **next** is part of the MH (Message Handling) package and can be used with other MH and AIX commands.

The **next** command links to the **show** program and passes **show** its flags and attributes.

**Note:** If you link to **next** and call that link something other than **next**, your link will function like the **show** command, rather than like the **next** command.

The **show** command invokes a program to perform the listing. The system default is **/bin/pg**. You can define your own default with the **showproc:** entry in **\$HOME/.mh-profile**. If you set **showproc:** entry to **mhl**, **show** calls an internal **mhl** routine instead of the **mhl** command. You can also specify the program to perform a listing in the **cmdstring** of the **-showproc** flag.

The **show** command passes any flags that it does not recognize to the program performing the listing. Thus, you can specify flags for the listing program, as well as the flags described in this command section.

## Flags

|                            |                                                                                                                              |
|----------------------------|------------------------------------------------------------------------------------------------------------------------------|
| <b>+folder</b>             | Specifies the folder that contains the message you want to show.                                                             |
| <b>-header</b>             | Displays a one-line description of the message being shown. The description includes the folder name and the message number. |
| <b>-help</b>               | Displays help information for the command.                                                                                   |
| <b>-noheader</b>           | Does not display a one-line description of each message being shown.                                                         |
| <b>-noshowproc</b>         | Uses <b>/bin/cat</b> to perform the listing.                                                                                 |
| <b>-showproc cmdstring</b> | Uses the specified command string to perform the listing.                                                                    |

## Profile Entries

|                        |                                              |
|------------------------|----------------------------------------------|
| <b>Current-Folder:</b> | Sets your default current folder.            |
| <b>Path:</b>           | Specifies your <i>user_mh_directory</i> .    |
| <b>showproc:</b>       | Specifies the program used to show messages. |

## Files

|                                 |                      |
|---------------------------------|----------------------|
| <code>\$HOME/.mh-profile</code> | The MH user profile. |
|---------------------------------|----------------------|

## Related Information

Other MH commands: “**prev**” on page 765, “**show**” on page 942.

The **mh-format**, **mh-mail**, and **mh-profile** files in *AIX Operating System Technical Reference*.

“Overview of the Message Handling Package” in *Managing the AIX Operating System*.



## nfsd

---

## nfsd

---

### Purpose

Starts the daemons that handle client Network File System requests.

### Syntax

```
nfsd _____ nservers_____
```

OL805488

### Description

The **nfsd** command starts the daemons that handle client file system requests.

The *nservers* parameter specifies the number of file system request daemons to start. Assign the number based on the load expected on a server. Four is an average load number.

When a file opened by a client is unlinked, a new file is created by the client. The new file is in the form **.nfsxxx**. When the open file is closed, the **.nfsxxx** file is removed.

**Note:** If the client crashes before the file can be closed, the **.nfsxxx** file is not removed.

---

#### Japanese Language Support Information

If Japanese Language Support is installed on your system, this command is not available.

---

### Files

**.nfsxxx** Client machine's pointer to an open file that has been unlinked.  
**/etc/rc.nfs**

### Related Information

The following command, "**biod**" on page 114.

The IBM AIX/RT Network File System section in *Managing the AIX Operating System*.

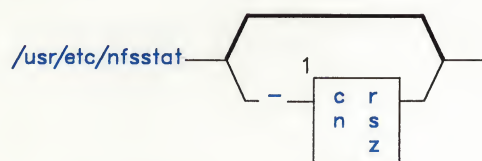
# nfsstat

---

## Purpose

Displays Network File System statistics.

## Syntax



<sup>1</sup> Do not put a blank between these items.

OL805489

OL805308

## Description

The **nfsstat** command displays statistical information about NFS (Network File System) and RPC (Remote Procedure Call).

If you have superuser authority, you can also use **nfsstat** to reinitialize the NFS and RPC statistical information.

The default for **nfsstat** is **nfsstat -csnr**. If you use it without flags, information for NFS and RPC is displayed and reinitialization does not occur.

---

### Japanese Language Support Information

If Japanese Language Support is installed on your system, this command is not available.

---

## Flags

Use the following flags for displaying and reinitializing the statistics for specific areas:

- c Displays NFS and RPC client information.
- cn Displays NFS client information only.
- cr Displays RPC client information only.

## nfsstat

---

- n** Displays NFS client and server information.
- r** Displays RPC client and server information.
- s** Displays NFS and RPC server information.
- sn** Displays NFS server information only.
- sr** Displays RPC server information only.
- z** Reinitializes statistics to zero. Only users with superuser authority can reinitialize statistics.

After displaying certain sets of the statistics, you can reinitialize them by using the flag that identifies the area followed by the **-z** flag. For example, you would use the flag combination **-cnz** to reinitialize NFS client statistics.

## Files

/unix        System name list  
/dev/kmem    Kernel memory

## Related Information

The following command: “**rstatd**” on page 847.



# nice

## Purpose

Runs a command at a different priority.

## Syntax

```
nice -[number] cmdstring
```

<sup>1</sup> Maximum increment is 19.

OL805200

## Description

The **nice** command lets you run the specified *command* at a lower priority. The value of *number* can range from 1 to 19, with 19 being the lowest priority. The default value of *number* is 10.

If you have superuser authority, you can run commands at a higher priority by specifying *number* as a negative number, such as --10.

## Examples

1. To run a command at low priority:

```
nice cc -c *.c
```

This runs the command `cc -c *.c` at low priority. Note that this does not run the command in the background. Your work station is not available for doing other things.

2. To run a low priority command in the background:

```
nice cc -c *.c &
```

This runs the command `cc -c *.c` at low priority in the background. Your work station is free so that you can run other commands while `cc` is running. See page 914 for details about starting background processes with `&`.

3. To specify a very low priority:

```
nice -15 cc -c *.c &
```

This runs **cc** in the background at a priority that is even lower than the default priority set by **nice**.

4. To specify a very high priority:

```
nice --10 wall <<end  
System shutdown in 2 minutes!  
end
```

This runs **wall** at a higher priority than all user processes. Doing this slows down everything else running on the system. If you do not have superuser authority when you run this command, then the **wall** command runs at the normal priority.

The <<end and end define a “Here Document,” which uses the text entered before the end line as standard input for the command. For more details, see “Inline Input Documents” on page 928.

## Related Information

The following commands: “**cs**h” on page 225 and “**no**hup” on page 707.

**Note:** The **cs**h command contains a built-in subcommand named **nice**. The command and subcommand do not necessarily work the same way. For information on the subcommand, see the **cs**h command.

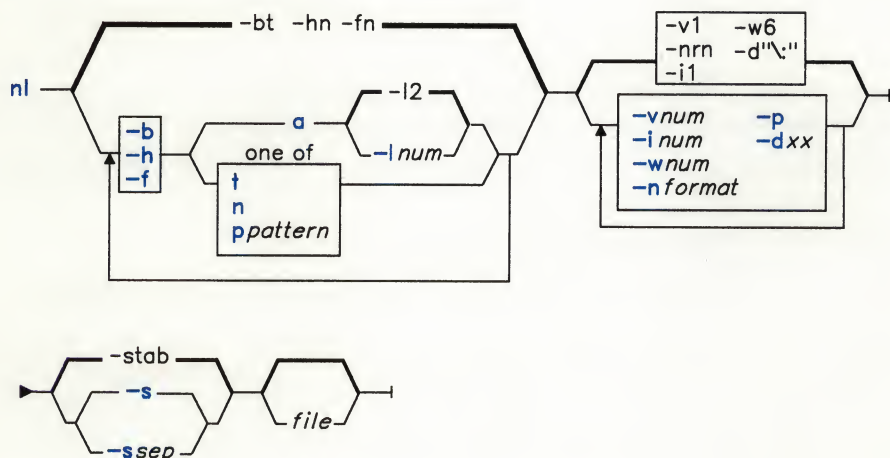
The **nice** system call in *AIX Operating System Technical Reference*.

## nl

## Purpose

Numbers lines in a file.

## Syntax



OL805386

## Description

The `nl` command reads *files* (standard input by default), numbers the lines in the input, and writes the numbered lines to standard output. In the output, `nl` numbers the lines on the left according to the flags you specify on the command line.

The input test must be written in logical pages. Each logical page has a header, a body, and a footer section (you can have empty sections). Unless you use the `-p` flag, `nl` resets the line numbers at the start of each logical page. You can set line numbering flags independently for the header, body, and footer sections (for example, no numbering of header and footer lines while numbering text lines only in the body).



Signal the start of logical page sections with lines in *file* that contain nothing but the following delimiter characters:

| <i>Line contents</i> | <i>Start of</i> |
|----------------------|-----------------|
| \:\::                | Header          |
| \::                  | Body            |
| \:                   | Footer          |

You can name only one file on the command line. You can list the flags and the file name in any order.

## Flags

All the parameters are set by default. Use the following flags to change these default settings. Except for the **-s** flag, enter a flag without a parameter to see its default value.

- btype** Chooses which body section lines to number. The recognized *types* are:
- |                 |                                                                    |
|-----------------|--------------------------------------------------------------------|
| <b>a</b>        | Numbers all lines.                                                 |
| <b>t</b>        | Does not number blank lines (default).                             |
| <b>n</b>        | Does not number any lines.                                         |
| <b>ppattern</b> | Numbers only those lines containing the specified <i>pattern</i> . |
- dxx** Uses *xx* as the delimiters for the start of a logical page section. The default characters are \: (backslash followed by a colon). You may specify two ASCII characters, two 1-byte extended characters, or one extended character. If you enter only one 1-byte character after **-d**, the second character remains the default (colon). If you want to use a backslash as a delimiter, enter two backslashes (\\).
- ftype** Chooses which logical page footer lines to number. The *types* recognized are the same as in **-btype**. The default *type* is **n** (no lines numbered).
- htype** Chooses which logical page header lines to number. The *types* recognized are the same as in **-btype**. The default *type* is **n** (no lines numbered).
- inum** Increments logical page line numbers by *num*. The default value of *num* is 1.
- lnum** Uses *num* as the number of blank lines to count as one. For example, **-l3** will only number the third adjacent blank. The default value of *num* is 2. This flag can only be used in documents where the **-ba** flag is used.
- nformat** Uses *format* as the line numbering format. Recognized formats are:
- |           |                                                      |
|-----------|------------------------------------------------------|
| <b>ln</b> | Left justified, leading zeros suppressed.            |
| <b>rn</b> | Right justified, leading zeros suppressed (default). |
| <b>rz</b> | Right justified, leading zeros kept.                 |
- p** Does not restart numbering at logical page delimiters.

- s[sep]** Separates the text from its line number by the *sep* character. The default value of *sep* is a tab character. If you enter **-s** without a parameter, there is no separation between the line number and its text.
- vnum** Sets the initial logical page line number to *num*, (1 by default).
- wnum** Uses *num* as the number of characters in the line number. The default value of *num* is 6.

## Examples

1. To number only the nonblank lines:

```
nl chap1
```

This displays a numbered listing of chap1, numbering only the nonblank lines in the body sections. If chap1 contains no `\: \ + :`, `\: \ + :`, or `\: \ + :` delimiters, then the entire file is considered the body.

2. To number all lines:

```
nl -ba chap1
```

This numbers all the lines in the body sections, including blank lines. This form of the **nl** command is adequate for most uses.

3. To specify a different line number format:

```
nl -i10 -nrz -s:: -v10 -w4 chap1
```

This numbers the lines of chap1 starting with ten (**-v10**) and counting by tens (**-i10**). It displays four digits for each number (**-w4**), including leading zeros (**-nrz**). The line numbers are separated from the text by two colons (**-s::**).

For example, if chap1 contains the text:

```
A not-so-important
note to remember:

You can't kill time
without injuring eternity.
```

then the numbered listing is:

```
0010::A not-so-important
0020::note to remember:

0030::You can't kill time
0040::without injuring eternity.
```

Note that the blank line was not numbered. To do this, use the **-ba** flag as shown in Example 2.

nl

---

## Related Information

The following command: “**pr**” on page 761.

“Overview of International Character Support” in *Managing the AIX Operating System*.



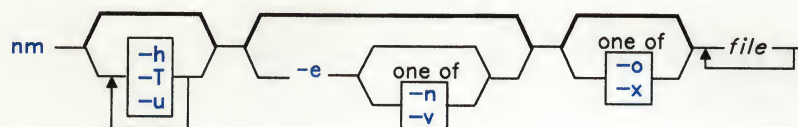
---

**nm**


---

**Purpose**

Displays the symbol table of an object file.

**Syntax**

OL805202

**Description**

The **nm** command writes the symbol table of each specified object *file* to standard output. *file* can be a single relocatable or absolute common object file or an archive library of relocatable or absolute common object files. **nm** displays the following information for each symbol:

|                |                                                                                                                                                                                                                                                                                                         |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Name</b>    | The name of the symbol.                                                                                                                                                                                                                                                                                 |
| <b>Value</b>   | Its value expressed as an offset or an address depending on its storage class.                                                                                                                                                                                                                          |
| <b>Class</b>   | Its storage class.                                                                                                                                                                                                                                                                                      |
| <b>Type</b>    | Its type and derived type. If the symbol refers to a structure or a union, the structure or union tag follows the type declaration. If the symbol is an array, the array dimensions follow the type. Note that you must have compiled the object file with <b>cc -g</b> for this information to appear. |
| <b>Size</b>    | Its size in bytes, if available. Note that you must have compiled the object file with <b>cc -g</b> for this information to appear.                                                                                                                                                                     |
| <b>Line</b>    | The source line number at which it is defined, if available. Note that you must have compiled the object file with <b>cc -g</b> for this information to appear.                                                                                                                                         |
| <b>Section</b> | For static and external storage classes, the object file section containing the symbol.                                                                                                                                                                                                                 |

### Flag

- e Displays only static and external symbols.
- h Does not display output header data.
- n Sorts external symbols by name before displaying them. Use this flag only in conjunction with the -e flag.
- o Displays a symbol's value and size as an octal rather than a decimal number.
- T Truncates every name that would otherwise overflow its column, making the last character displayed in the name an asterisk. By default, **nm** displays the entire name of the symbols listed, and a name that is longer than the width of the column set aside for it causes every column after the name to be misaligned.
- u Displays only undefined symbols.
- v Sorts external symbols by value before displaying them. Use this flag only in conjunction with the -e flag.
- x Displays a symbol's value and size as a hexadecimal rather than a decimal number.

### Files

a.out Default input file.

### Related Information

The following commands: “**ar**” on page 55, “**as**” on page 61, “**backup**” on page 88, “**cc**” on page 140, “**ld**” on page 557, “**size**” on page 949, and “**strip**” on page 1017.

The **a.out** and **ar** files in *AIX Operating System Technical Reference*.

## nohup

---

### Purpose

Runs a command without hangups and quits.

### Syntax

`nohup` — *command* —

OL805203

### Description

The **nohup** command runs *command*, ignoring all hangups and **QUIT** signals. You can use this command to run programs in the background after you log off of the system. To run a **nohup** command in the background, add an **&** to the end of the command.

If **nohup** output is redirected to a terminal or is not redirected at all, the output goes to the file **nohup.out**. If **nohup.out** is not writable in the current directory, the output is redirected to **\$HOME/nohup.out**.

The syntax of this command ignores quits and hangups for only one *command*. If you want to apply **nohup** to a pipeline or list of commands, you can put the pipeline or list in a shell script file. Then you can run **sh** as the *command* using the format: `nohup sh file`. You can also assign the shell file execute permission and run it as the command in the form: `nohup file`.

### Examples

1. To leave a command running after you log off:

```
nohup find / -print &
```

Shortly after you enter this, the following is displayed:

```
670  
$ Sending output to nohup.out
```



## nohup

---

The number will probably be different when you use this command. It is the ID of the background process started by & (ampersand). (See page 914 about starting background processes with &.) The \$ (dollar sign) is your shell prompt. Sending output . . . is a message from **nohup** telling you that it is storing the output from the `.find` command in the file `nohup.out`. You can log off after you see these messages, even if the `find` command has not finished yet.

2. To do the same, but redirect the standard output to a different file:

```
nohup find / -print >filenames &
```

This runs the `find` command and stores its output in a file named `filenames`. Now only the process ID and your prompt are displayed:

```
677  
$
```

Wait for a second or two before logging off, because the **nohup** command takes a moment to start the *command* you specified. If you log off too quickly, your *command* may not run at all. Once your *command* has started, logging off will not affect it.

3. To run more than one command, use a shell procedure. For example, if you write the shell procedure:

```
neqn math1 ] nroff > fmath1
```

and name it `nnfmath1`, you can run **nohup** for all of the commands in `nnfmath1` with the command:

```
nohup sh nnmath1
```

If you assign `nnfmath1` execute permission, you can obtain the same results by issuing the command:

```
nohup nnmath1
```

To run this command in the background, enter the command:

```
nohup nnmath1 &
```

## Related Information

The following commands: “**csh**” on page 225, “**nice**” on page 699, and “**sh**” on page 913.

**Note:** The **csh** command contains a built-in subcommand named **nohup**. The command and subcommand do not necessarily work the same way. For information on the subcommand, see the **csh** command.

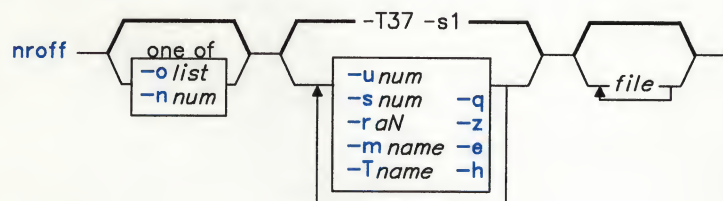
The **signal** system call in *AIX Operating System Technical Reference*.

# nroff, troff

## Purpose

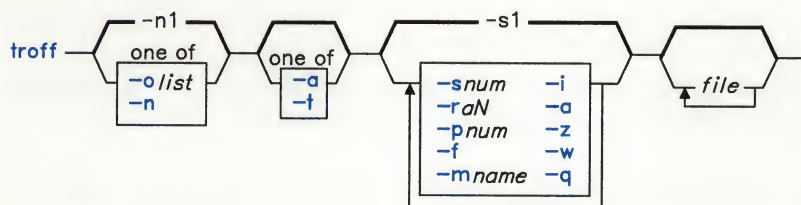
Formats text for printing devices.

## Syntax



OL805204

**troff** — **-b** —



OL805368

## Description

A complete list of **nroff** and **troff** requests, escape sequences and number registers begins on page 712. See *Text Formatting Guide* for a complete list of the naming conventions for the non-ASCII special characters and for information on writing text suitable for processing by **troff** or **nroff**.

### nroff, troff

The **nroff** command reads *files* (standard input by default), formats the text in its input for printing, and writes to standard output. **nroff** formats text for line printers and other *printing devices*, excluding phototypesetters. An input file name of - (minus) indicates standard input.

### troff

The **troff** command formats text in the input *files* (or standard input by default) for a phototypesetter, and writes its output to standard output. It is similar to the **nroff** command. An input file name of - (minus) indicates standard input.

### nroff and troff Flags:

- i Reads standard input after the input files.
- mname Adds */usr/lib/tmac/tmac.name* to the beginning of the list of input file names.
- num Numbers the first printed page *num*. Do not use this flag with **-olist**.
- olist Prints only pages with page numbers appearing in *list* which consist of a comma-separated list of page numbers and ranges. A range of *A-B* means print pages *A* through *B*; an initial *-A* means print from the beginning to page *A*; and a final *A-* means print from page *A* to the end.

**Note:** When this flag is used in a pipeline (for example, with **cw**, **eqn**, or **tbl**) it may cause a broken pipe diagnostic if the last page in the document is not specified in *list*.
- q Invokes the simultaneous input/output mode of the **.rd** request. **nroff** echoes the **.rd** prompt, but does not echo your input. When you enter two consecutive new-line characters, normal output is resumed.
- raN Sets register *a* to *N*. *a* must have a one-character name. This is useful for automatic numbering of sections, paragraphs, lines, and so forth.
- snum Stops every *num* pages (the default is 1). **nroff** or **troff** will halt every *num* pages to allow paper loading or changing and will resume upon receipt of a line-feed or new-line character. This flag does not work in pipelines. When **nroff** halts between pages, an ASCII BEL character is sent to the printing device.
- z Suppresses the formatted output. Prints only messages generated by **.tm** (work station message) requests.



### ***nroff* Flags:**

- e Produces equally spaced words in adjusted lines, using the full resolution of the printing device.
- h Uses tab characters during horizontal spacing. Tab settings are assumed to be every eight spaces.
- T*name* Prepares the output for the specified printing device. Known *names* are:
  - 37 TELETYPE Model 37 work station (default)
  - tn300 GE TermiNet 300 or any work station without half-line capability
  - 300s DASI 300s
  - 300 DASI 300
  - 450 DASI 450
  - lp Any ASCII line printer
  - 382 DCT-382
  - 4000A Trendata 4000A
  - 832 Anderson Jacobson 832
  - X any EBCDIC printer
  - 2631 Hewlett Packard 2631 line printer. Use a *name* of **2631-c** to get compressed print. Use **2631-e** to get expanded print.
- u[*num*] Sets the number of character overstrikes for boldface to *num* or to zero if *num* is not specified.

### ***troff* Flags:**

- a Sends a printable ASCII approximation of the output to standard output.
- b Reports whether the phototypesetter is busy or available. No text processing is done.
- f Does not feed out paper and stop the phototypesetter at the end of the run.
- p*num* Prints all characters in the point size specified by *num*. Smaller point sizes may reduce the printing time.
- t Directs output without modification to standard output instead of the phototypesetter.
- w Waits until phototypesetter is available if it is currently busy.

## nroff

---

### nroff and troff Requests

| Request Form       | Function -- Font and Character Size Control                                          |
|--------------------|--------------------------------------------------------------------------------------|
| <b>.ps</b> $\pm N$ | Change point size by $N$ points. Also, for <b>troff</b> only, $\backslash s \pm N$ . |
| <b>.ss</b> $N$     | Space-character size set to $N/36$ em ( <b>troff</b> ) only.                         |
| <b>.cs</b> $F N M$ | Constant character space (width) mode (font $F$ ) ( <b>troff</b> only).              |
| <b>.bd</b> $F N$   | Embolden font $F$ by $N$ units ( <b>troff</b> only).                                 |
| <b>.bd</b> $S F N$ | Embolden Special Font when current font is $F$ ( <b>troff</b> only).                 |
| <b>.ft</b> $F$     | Change to font $F$ .                                                                 |
| <b>.fp</b> $N F$   | Mount font $F$ on position $N$ (1-4).                                                |

| Request Form       | Function -- Page Control                       |
|--------------------|------------------------------------------------|
| <b>.pl</b> $\pm N$ | Change page length by $N$ .                    |
| <b>.bp</b> $\pm N$ | Eject current page; next page number is $N$ .  |
| <b>.pn</b> $N$     | Next page number is $N$ .                      |
| <b>.po</b> $\pm N$ | Page offset = $N$ .                            |
| <b>.ne</b> $N$     | Need $N$ vertical space.                       |
| <b>.mk</b> $R$     | Mark current vertical place in register $R$ .  |
| <b>.rt</b> $\pm N$ | Return (upward only) to marked vertical place. |

| Request Form     | Function -- Text Filling, Adjusting, and Centering |
|------------------|----------------------------------------------------|
| <b>.br</b>       | Break.                                             |
| <b>.fi</b>       | Fill subsequent output lines.                      |
| <b>.nf</b>       | No filling or adjusting of output lines.           |
| <b>.ad</b> $[c]$ | Adjust output lines with mode $c$ .                |
| <b>.na</b>       | Do not adjust output lines.                        |
| <b>.ce</b> $N$   | Center the following $N$ lines.                    |



| Request Form        | Function -- Vertical Spacing                                     |
|---------------------|------------------------------------------------------------------|
| <b>.vs</b> <i>N</i> | Set vertical base-line spacing to <i>N</i> .                     |
| <b>.ls</b> <i>N</i> | Output <i>N</i> -1 base-line spaces after each text output line. |
| <b>.sp</b> <i>N</i> | Space vertical distance <i>N</i> in either direction.            |
| <b>.sv</b> <i>N</i> | Save vertical distance <i>N</i> .                                |
| <b>.os</b>          | Output saved vertical space.                                     |
| <b>.ns</b>          | Turn no-space mode on.                                           |
| <b>.rs</b>          | Restore spacing; turn no-space mode off.                         |

| Request Form       | Function -- Line Length and Indenting            |
|--------------------|--------------------------------------------------|
| <b>.li</b> $\pm N$ | Change line length by <i>N</i> .                 |
| <b>.in</b> $\pm N$ | Change indenting by <i>N</i> .                   |
| <b>.ti</b> $\pm N$ | Change the indent on the next line by <i>N</i> . |

| Request Form                | Function -- Macros, Strings, Diversion, and Position Traps          |
|-----------------------------|---------------------------------------------------------------------|
| <b>.de</b> <i>xx yy</i>     | Define or redefine macro <i>xx</i> ; end at call of <i>yy</i> .     |
| <b>.am</b> <i>xx yy</i>     | Append to a macro.                                                  |
| <b>.ds</b> <i>xx string</i> | Define a string <i>xx</i> containing <i>string</i> .                |
| <b>.as</b> <i>xx string</i> | Append <i>string</i> to string <i>xx</i> .                          |
| <b>.rm</b> <i>xx</i>        | Remove request, macro, or string named <i>xx</i> .                  |
| <b>.rn</b> <i>xx yy</i>     | Rename request, macro, or string <i>xx</i> to <i>yy</i> .           |
| <b>.di</b> <i>xx</i>        | Divert output to macro <i>xx</i> .                                  |
| <b>.da</b> <i>xx</i>        | Divert and append to <i>xx</i> .                                    |
| <b>.wh</b> <i>N xx</i>      | Set location trap; negative is with respect to the end of the page. |
| <b>.ch</b> <i>xx N</i>      | Change trap location.                                               |
| <b>.dt</b> <i>N xx</i>      | Set a diversion trap.                                               |
| <b>.it</b> <i>N xx</i>      | Set an input line trap.                                             |
| <b>.em</b> <i>xx</i>        | End macro is <i>xx</i> .                                            |



## nroff

| Request Form                         | Function -- Number Registers                                           |
|--------------------------------------|------------------------------------------------------------------------|
| <b>.nr</b> <i>R</i> $\pm$ <i>N M</i> | Define and set number register <i>R</i> ; auto-increment by <i>M</i> . |
| <b>.af</b> <i>R c</i>                | Assign format to register <i>R</i> ( <i>c</i> = 1, i, I, a, A).        |
| <b>.rr</b> <i>R</i>                  | Remove register <i>R</i> .                                             |

| Request Form               | Function -- Tabs, Leaders, and Fields                                 |
|----------------------------|-----------------------------------------------------------------------|
| <b>.ta</b> <i>Nt</i> . . . | Tab settings; left type, unless <i>t</i> = R (right) or C (centered). |
| <b>.tc</b> <i>c</i>        | Tab repetition character.                                             |
| <b>.lc</b> <i>c</i>        | Leader repetition character.                                          |
| <b>.fc</b> <i>a b</i>      | Set field delimiter <i>a</i> and pad character <i>b</i> .             |

| Request Form                 | Function -- Input/Output Conventions and Character Translations                       |
|------------------------------|---------------------------------------------------------------------------------------|
| <b>.ec</b> <i>c</i>          | Set escape character.                                                                 |
| <b>.eo</b>                   | Turn off escape character mechanism.                                                  |
| <b>.lg</b> <i>N</i>          | Ligature on if <i>N</i> > 0.                                                          |
| <b>.ul</b> <i>N</i>          | Underline in <b>nroff</b> or italicize in <b>troff</b> the next <i>N</i> input lines. |
| <b>.cu</b> <i>N</i>          | Continuous underline in <b>nroff</b> . Acts like <b>.ul</b> in <b>troff</b> .         |
| <b>.uf</b> <i>F</i>          | Underline font set to <i>F</i> (to be switched to by <b>.ul</b> ).                    |
| <b>.cc</b> <i>c</i>          | Set control character to <i>c</i> .                                                   |
| <b>.c2</b> <i>c</i>          | Set no-break control character to <i>c</i> .                                          |
| <b>.tr</b> <i>abcd</i> . . . | Translates <i>a</i> to <i>b</i> , and so on, on output.                               |

| Request Form                    | Function -- Hyphenation                    |
|---------------------------------|--------------------------------------------|
| <b>.nh</b>                      | No hyphenation.                            |
| <b>.hy</b> <i>H</i>             | Hyphenate; <i>N</i> = mode.                |
| <b>.hc</b> <i>c</i>             | Hyphenation indicator character <i>c</i> . |
| <b>.wc</b><br><i>word</i> . . . | Exception words.                           |

| Request Form                                       | Function -- Three Part Titles |
|----------------------------------------------------|-------------------------------|
| .tl ' <i>left</i> ' <i>center</i> ' <i>right</i> ' | Three part title.             |
| .pc <i>c</i>                                       | Page number character.        |
| .lt $\pm N$                                        | Length of title.              |

| Request Form      | Function -- Output Line Numbering      |
|-------------------|----------------------------------------|
| .nm $\pm N M S I$ | Number mode on or off, set parameters. |
| .nm <i>N</i>      | Do not number next <i>N</i> lines.     |

| Request Form                                              | Function -- Conditional Acceptance of Input                                                                         |
|-----------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------|
| .if <i>c.anything</i>                                     | If condition <i>c</i> is true, accept <i>anything</i> as input. For multiple lines, use <code>\{anything\}</code> . |
| .if ! <i>c anything</i>                                   | If condition <i>c</i> is false, accept <i>anything</i> as input.                                                    |
| .if <i>N anything</i>                                     | If expression $N > 0$ , accept <i>anything</i> as input.                                                            |
| .if ! <i>N anything</i>                                   | If expression $N \leq 0$ , accept <i>anything</i> as input.                                                         |
| .if ' <i>string1</i> ' <i>string2</i> ' <i>anything</i>   | If <i>string1</i> is identical to <i>string2</i> , accept <i>anything</i> as input.                                 |
| .if '! <i>string1</i> ' <i>string2</i> .' <i>anything</i> | If <i>string1</i> is not identical to <i>string2</i> , accept <i>anything</i> as input.                             |
| .ie <i>c anything</i>                                     | If part of if-else; can take all forms of if above.                                                                 |
| .el <i>anything</i>                                       | Else part of if-else.                                                                                               |

| Request Form | Function -- Environment Switching |
|--------------|-----------------------------------|
| .ev <i>N</i> | Environment switched (push down). |

| Request Form      | Function -- Insertions from Standard Input |
|-------------------|--------------------------------------------|
| .rd <i>prompt</i> | Read insertion.                            |
| .ex               | Exit from nroff or troff.                  |



## nroff

| Request Form              | Function -- Input/Output File Switching             |
|---------------------------|-----------------------------------------------------|
| <b>.so</b> <i>file</i>    | Switch source file (push down).                     |
| <b>.nx</b> <i>file</i>    | Next file.                                          |
| <b>.pi</b> <i>program</i> | Pipe output to <i>program</i> ( <b>nroff</b> only). |

| Request Form               | Function -- Miscellaneous                                                                                          |
|----------------------------|--------------------------------------------------------------------------------------------------------------------|
| <b>.mc</b> <i>c N</i>      | Set margin character <i>c</i> and separation <i>N</i> .                                                            |
| <b>.tm</b> <i>string</i>   | Print <i>string</i> on standard error output.                                                                      |
| <b>.ig</b> <i>yy</i>       | Ignore until call of <i>yy</i> .                                                                                   |
| <b>.pm</b> <i>t</i>        | Print macro names and sizes; if <i>t</i> is present, print only the total of sizes.                                |
| <b>.fl</b>                 | Flush output buffer.                                                                                               |
| <b>.ab</b> [ <i>text</i> ] | Prints <i>text</i> on standard error output and stops output. User abort is printed if no <i>text</i> is included. |
| <b>! cmd parms</b>         | Runs the AIX command <i>cmd</i> and interpolates at that point. The standard input for <i>cmd</i> is closed.       |

## Escape Sequences for Characters, Indicators, and Functions

| Escape Sequence | Meaning                                                      |
|-----------------|--------------------------------------------------------------|
| <b>\\</b>       | Prevents or delays interpretation of <b>\</b> .              |
| <b>\e</b>       | Printable version of the current escape character.           |
| <b>\.</b>       | Acute accent; equivalent to <b>\(aa</b> .                    |
| <b>\`</b>       | Grave accent; equivalent to <b>\(ga</b> .                    |
| <b>\-</b>       | Minus sign in the current font.                              |
| <b>\.</b>       | Dot.                                                         |
| <b>\(space)</b> | Unpaddable space-size character.                             |
| <b>\0</b>       | Digit width space.                                           |
| <b>\\</b>       | 1/6 em narrow space character (zero width in <b>nroff</b> ). |



| Escape Sequence | Meaning                                                                                                           |
|-----------------|-------------------------------------------------------------------------------------------------------------------|
| \^              | 1/12 em half-narrow space character (zero width in <b>nroff</b> ).                                                |
| \&              | Non-printing, zero-width character.                                                                               |
| \!              | Transparent line indicator.                                                                                       |
| \\$N            | Interpolate argument $1 \leq N \leq 9$ .                                                                          |
| \%              | Default optional hyphenation character.                                                                           |
| \(xx            | Character named <i>xx</i> .                                                                                       |
| \\*x, \*(xx     | Interpolate string <i>x</i> or <i>xx</i> .                                                                        |
| \a              | Non-interpreted leader character.                                                                                 |
| \b'abc . . . '  | Bracket building function.                                                                                        |
| \c              | Interrupt text processing (continue word across input line break).                                                |
| \d              | Forward (down) 1/2 em vertical motion (1/2 line in <b>nroff</b> ).                                                |
| \fx, \f(xx, \fN | Change to font <i>N</i> named <i>x</i> or <i>xx</i> , or font position <i>N</i> .                                 |
| \gx, \g(xx      | Return the format of register <i>x</i> or <i>xx</i> . Return nothing if the register has not yet been referenced. |
| \h'N'           | Local horizontal motion; move right <i>N</i> (negative left).                                                     |
| \jx, \jxx       | Mark in register <i>x</i> or <i>xx</i> the current horizontal position on the output line.                        |
| \kx             | Mark horizontal input place in register <i>x</i> .                                                                |
| \l'N[c]'        | Horizontal line drawing function.                                                                                 |
| \L'N[c]'        | Vertical line drawing function.                                                                                   |
| \nx, \(xx       | Interpolate number register <i>x</i> or <i>xx</i> .                                                               |
| \o'abc . . . '  | Overstrike characters <i>a</i> , <i>b</i> , <i>c</i> , . . . .                                                    |
| \p              | Break and spread output line.                                                                                     |
| \r              | Reverse 1 em vertical motion (reverse line in <b>nroff</b> ).                                                     |
| \sN, \s $\pm$ N | Point-size change function.                                                                                       |
| \t              | Non-interpreted horizontal tab.                                                                                   |
| \u              | Reverse (up) 1/2 em vertical motion (1/2 line in <b>nroff</b> ).                                                  |
| \v'N'           | Local vertical motion ; move down <i>N</i> (negative up).                                                         |
| \w'string'      | Interpolate width of <i>string</i> .                                                                              |

| Escape Sequence          | Meaning                                                      |
|--------------------------|--------------------------------------------------------------|
| <code>\x'N'</code>       | Extra line-space function (negative before; positive after). |
| <code>\zc</code>         | Print <i>c</i> with zero width without spacing.              |
| <code>\{</code>          | Begin conditional input.                                     |
| <code>\}</code>          | End conditional input.                                       |
| <code>\(new line)</code> | Concealed new line.                                          |
| <code>\X</code>          | <i>X</i> , any character not listed above.                   |

### Predefined General Number Registers

| Register Name   | Description                                                     |
|-----------------|-----------------------------------------------------------------|
| <code>%</code>  | Current page number.                                            |
| <code>ct</code> | Character width type (set by width function).                   |
| <code>dl</code> | Maximum width of last completed diversion.                      |
| <code>dn</code> | Height (vertical size) of last completed diversion.             |
| <code>dw</code> | Current day of the week (1 = Sunday . . . 7 = Saturday).        |
| <code>dy</code> | Current day of the month (1-31).                                |
| <code>hp</code> | Current horizontal place on the input line.                     |
| <code>ln</code> | Output line number.                                             |
| <code>mo</code> | Current month (1-12).                                           |
| <code>nl</code> | Vertical position of last printed text baseline.                |
| <code>sb</code> | Depth of string below base line (generated by width function).  |
| <code>st</code> | Height of string above base line (generated by width function). |
| <code>yr</code> | Last two digits of current year.                                |



## Predefined Read-Only Number Registers

| Register Name | Meaning                                                                                                                               |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------|
| \$            | Number of arguments available at the current macro level.                                                                             |
| .A            | Set to 1 in <b>troff</b> if the <b>-a</b> flag is used; always 1 in <b>nroff</b> .                                                    |
| .F            | The name of the current input file.                                                                                                   |
| .H            | Available horizontal resolution in basic units.                                                                                       |
| .L            | Contains the current line spacing parameter.                                                                                          |
| .P            | Contains the value 1 if the current page is being printed, 0 otherwise.                                                               |
| .R            | The number of columns available.                                                                                                      |
| .T            | Set to 1 in <b>nroff</b> , if the <b>-T</b> flag is used; always 0 in <b>troff</b> .                                                  |
| .V            | Available vertical resolution in basic units.                                                                                         |
| .a            | Post-line extra line-space most recently utilized using <code>\s'N'</code> .                                                          |
| .b            | Emboldening factor of the current font.                                                                                               |
| .c            | Number of lines read from current input file, including <b>.so</b> files.                                                             |
| .d            | Current vertical place in current diversion; equal to <b>nl</b> if no diversion.                                                      |
| .f            | Current font as physical quadrant.                                                                                                    |
| .h            | Text base-line high-water mark on current page or diversion.                                                                          |
| .i            | Current indent.                                                                                                                       |
| .j            | Current adjustment mode and type.                                                                                                     |
| .k            | Contains the horizontal size of the text portion of the current, partially-collected output line, if any, in the current environment. |
| .l            | Current line length.                                                                                                                  |
| .n            | Length of text portion on previous output line.                                                                                       |
| .o            | Current page offset.                                                                                                                  |
| .p            | Current page length.                                                                                                                  |
| .s            | Current point size.                                                                                                                   |
| .t            | Distance to the next trap.                                                                                                            |
| .u            | Equal to 1 in fill mode; equal to 0 in no-fill mode.                                                                                  |



| Register Name | Meaning                              |
|---------------|--------------------------------------|
| .v            | Current vertical line spacing.       |
| .w            | Width of previous character.         |
| .x            | Reserved version-dependent register. |
| .y            | Reserved version-dependent register. |
| .z            | Name of current diversion.           |

## Files

|                      |                                                |
|----------------------|------------------------------------------------|
| /usr/lib/suftab      | Suffix hyphenation tables.                     |
| /tmp/ta\$#           | Temporary file.                                |
| /usr/lib/tmac/tmac.* | Standard macro files.                          |
| /usr/lib/macros/*    | Standard macro files.                          |
| /usr/lib/font/*      | Font width tables for <b>troff</b> .           |
| /usr/lib/term/*      | Work station driving tables for <b>nroff</b> . |

## Related Information

The following commands: “**col**” on page 179, “**cw, checkcw**” on page 275, “**eqn, neqn, checkeq**” on page 395, “**mm, checkmm**” on page 663, “**mmt, checkmm**” on page 666, “**greek**” on page 499, “**tbl**” on page 1053, and “**tc**” on page 1056.

The **mm** miscellaneous facility in *AIX Operating System Technical Reference*.

The discussion of **nroff** and **troff** in *Text Formatting Guide*.

## number

---

### Purpose

Displays the written form of a number.

### Syntax

`/usr/games/number —`

OL805229

### Description

The **number** game displays the written form of a number that it reads from standard input. The largest number it can translate accurately contains 66 digits.

The **number** game does not prompt you for a number. Once loaded, it simply waits for input. To exit the program, press INTERRUPT (**Alt-Pause**) or END OF FILE (**Ctrl-D**).

### Example

To display the written form of several numbers:

You: `/usr/games/number`

829

System: eight hundred twenty nine.

...

You: `12345678`

System: twelve million.

three hundred forty five thousand.

six hundred seventy eight.

...

You: **Ctrl-D**

**number**

---